brug din AMSTRAD

John Dewhirst James Ryan

brug din AMSTRAD

John Dewhirst James Ryan

teknisk forlag as

Brug din AMSTRAD
Originaltitel: A child's guide to the
Amstrad Micro
© 1984 Cambridge University Press
© 1985 dansk udgave: Teknisk Forlag A/S
Oversat fra engelsk af Kirsten Bredegaard

Forlagsredaktion: Tove From Jørgensen
Omslag: Mette Plesner
Tegninger: David Parkins
Sats og montage: Satsform, Åbyhøj
Tryk: Teknisk Forlag A/S, København

ISBN 87-571-0889-7

INDHOLD

Introduktion side **4**

Tastaturet **7**
Kontroltaster
Bogstavtaster
Taltaster
Symboltaster
Editeringstaster

Når du skal rigtigt i gang **21**
Datamaskinen som skrivemaskine
Datamaskinen som regnemaskine
Datamaskinens hukommelse
Lagring af information
Informationssøgning

Særlige muligheder **36**
Skift af skærmtilstand
Tegning på skærmen
Orientering på skærmen
Afmærkning af punkter
Farveskift
Komponering af musik

Programskrivning **46**
Dine egne programmer
Opgaver

Opslag **81**
Henvisninger
Editering og skrivning af programmer
Brug af skærmen
Frembringelse af toner
Ordfunktioner
Talfunktioner
Logiske udtryk
Lagring og indlæsning af programmer
Generelle ordrer og funktioner
Indtastning af danske karakterer

INTRODUKTION

Alle kapitlerne i bogen er skrevet af forskellige personer. De er hver især ekspert på deres område og giver mange nyttige råd. Læs omhyggeligt de gode råd, de giver i begyndelsen af hvert kapitel. Hvis du følger deres råd, vil du få meget lettere ved at arbejde med datamaskinen.

KAPITLER

1. Tastaturet

af Linda Tastesen – sekretær.
Fr. Tastesen viser dig, hvordan de forskellige taster virker, og hvor de sidder på tastaturet.

2. Når du skal rigtigt i gang

af S. T. Rømer – opdager.
S. T. Rømer viser dig, hvordan du kommer i gang med datamaskinen: hvordan man får den til at udskrive tal og ord, beregne svar og – til sidst – hvordan man lagrer information i dens hukommelse.

3. Særlige muligheder

af Frede Arvetrip – kunstner.
F. Arvetrip viser dig, hvordan man tegner figurer med farver på skærmen, og hvordan man får datamaskinen til at spille musik.

4. Programskrivning

af Prof. P. Nittengryn – planlægger.
Prof. P. Nittengryn viser dig, hvordan du afprøver og bliver bekendt med programmering for til sidst at kunne lave dine egne programmer.

5. Opslag

af B. O. Gorm – bibliotekar.
B. O. Gorm viser dig, hvordan man bruger de forskellige opslagsafsnit.

Kære læser.
Hej med dig. Vi er de folk, som har skrevet bogen, og du vil lære os bedre at kende efterhånden, som du læser den igennem. Når du møder os, skal vi nok præsentere os selv.

Bogen er skrevet for folk, som ikke ved noget om datamaskiner, men som gerne vil læse om dem på egen hånd. Den er ikke blot skrevet for at fortælle dig, hvad datamaskinen kan, men – hvad der er meget vigtigere – hvad du kan få maskinen til at gøre for dig.

Vi har samlet alle vore gode ideer, så det bliver let for dig at finde ud af, hvordan Amstrad Mikrodatamaten arbejder. Bogen er inddelt i fem kapitler, og vi har hver skrevet et. De første fire kapitler skal læses og gennemarbejdes i rækkefølge, mens det sidste er til opslag og kan bruges, hver gang der er noget, du vil undersøge nærmere.

God fornøjelse!

Linda Tastesen

Stærkodder Thorbjørn Rømer

Per Nittengryn

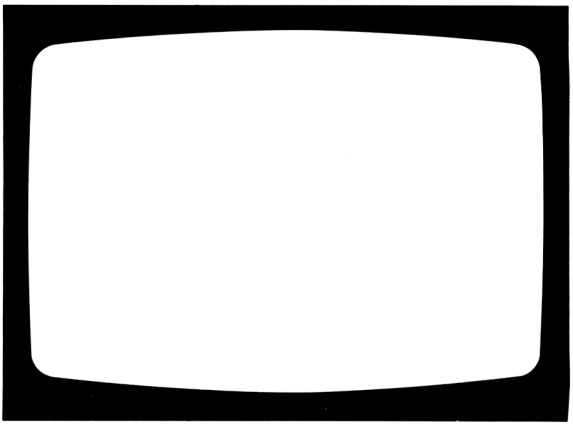
Frede Arvetrip

Bodil Oda Gorm

Hele bogen igennem vil du støde på forskellige slags figurer. De ting, som det er op til dig at afprøve – forskellige øvelser og forsøg – står i en "Prøv selv" kasse.
"Prøv selv" kassen ser sådan ud:

Prøv selv

Det, som kommer til at stå på skærmen, vises i en "Skærm" kasse.
"Skærm" kassen ser sådan ud:



De ting, som du bør skrive ned og huske, står i en "Skriv ned" kasse.
"Skriv ned" kassen ser sådan ud:

TASTATURET

LINDA TASTESEN

Mine råd

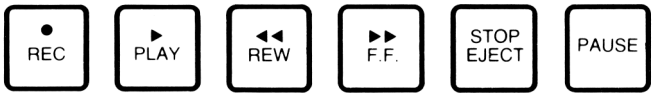
Vær ikke bange for datamaskinen. Jeg betragter den som en lidt anderledes skrivemaskine. Prøv dig frem. På den måde lærer du hurtigt, hvad den faktisk kan, og hvad den ikke kan. Når du afprøver de forskellige ting, sender maskinen besked tilbage til dig. Sådan taler den til dig. Den kan lave lyde, skifte farver på skærmen og gøre en masse andre ting. Derfor skal du nok bruge noget tid til at finde ud af alt det, som maskinen kan. Din første opgave bliver at lære noget om tastaturet. Du skal finde ud af, hvor alle tasterne sidder, og hvordan de hver især virker. Det er en god idé hele tiden at skrive tingene ned, som du lærer om datamaskinen.

Afprøv tingene undervejs samtidig med, at du læser om dem. Du skal ikke lade dig slå ud af de beskeder, maskinen sender til dig. Den forstår ikke, at du er begynder og bare prøver nogle ting af for dig selv. Når du går videre med resten af bogen, lærer du det sprog, datamaskinen forstår. Men ligesom når du lærer ethvert andet sprog, kan du kun lære lidt ad gangen.

Linda Tastesen

Der er 80 taster på Amstrad. De seks, der sidder længst til højre, styrer databåndet og fungerer på samme måde som knapperne på en almindelig båndoptager.

Amstrad Bruger-vejledning s. G2.1.



De andre 74 taster kan deles op i fem grupper:

Kontroltaster
Bogstavtaster
Taltaster
Symboltaster
Editeringstaster

Før du begynder at kigge nærmere på disse taster, er der nogle ting, du får brug for at vide.

1. Hvis du holder en tast nede, bliver den ved med at skrive det samme tegn. I stedet for et **a,** får du så **aaaaaaaa.**

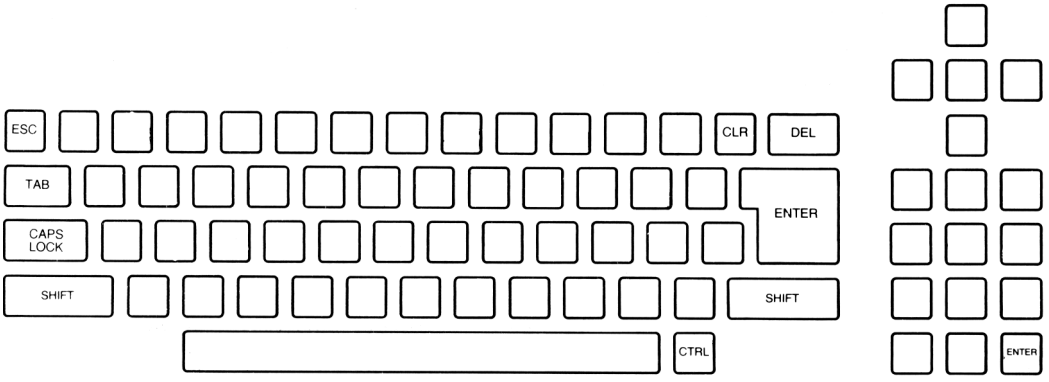
2. Du kan bruge **DEL** (som er en forkortelse for "delete": fjerne) og **CLR** (som står for "clear": slette) til at slette på skærmen. Det betyder, at du kan fjerne alle uønskede tegn ved at trykke på **DEL** og **CLR.** Men pas på! Disse taster fungerer ligesom de andre. Hvis du holder dem nede, kan du komme til at slette mere, end du havde tænkt dig.

Kontroltaster

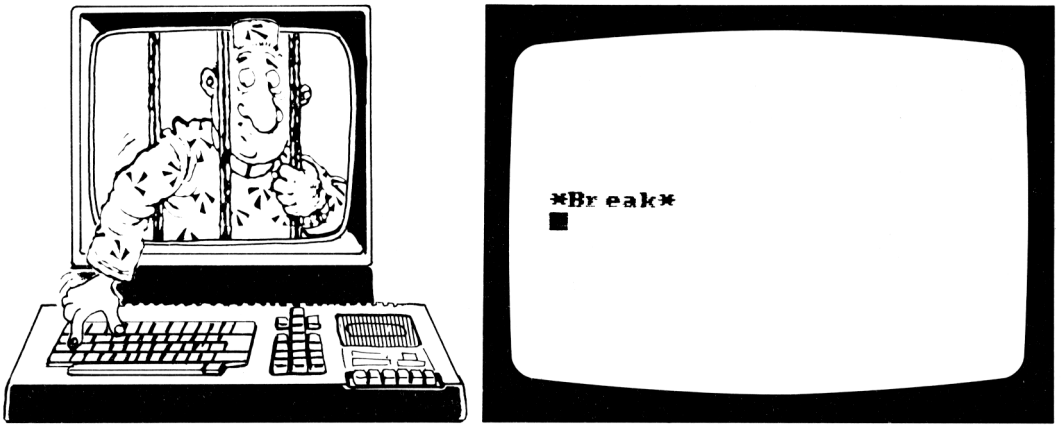
Der er ti kontroltaster.

ESC
TAB
CAPS LOCK
SHIFT (to taster)
DEL
CLR
ENTER (to taster)
CTRL

Amstrad Bruger-vejledning s. G2.1.



ESC er farvet rød. Den bruges til at afbryde det, datamaskinen er i gang med. Hvis maskinen bare står og venter på ordrer, kan du trykke på **ESC** og få denne besked på skærmen:



Hvis maskinen er ved at udføre et program, kan du ved at trykke en enkelt gang på **ESC** få den til at stoppe. Hvis du så trykker på en anden tast, går den videre igen derfra, hvor du afbrød den. Men hvis du trykker på **ESC** endnu en gang, holder maskinen helt op med det, den var i færd med. Den er da klar til, at du kan taste nye ordrer ind.

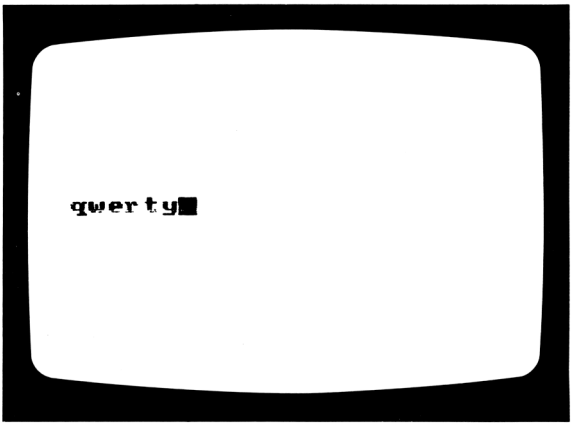
TAB er farvet grøn. Når du trykker på **TAB**, kommer der på skærmen en lille pil, som peger mod højre. Denne tast skriver ikke pilen flere gange, selv om du holder den nede.

CAPS LOCK Denne tast er farvet grøn. Når man trykker den ned en enkelt gang, skriver bogstavtasterne store bogstaver og symboltasterne de nederste symboler eller tal. Når man trykker på **CAPS LOCK** endnu en gang, slår man den fra, så bogstavtasterne igen skriver små bogstaver. Når man tænder for maskinen, er **CAPS LOCK** slået fra.

Prøv selv

Skriv **QWERTY**

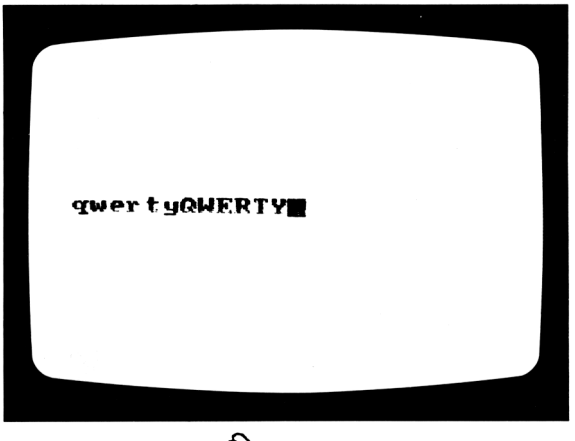
Sådan ser det ud på skærmen:



Prøv selv

Tryk på **CAPS LOCK** og skriv **QWERTY** igen. Prøv så at trykke på **CAPS LOCK** endnu en gang for igen at kunne skrive små bogstaver.

Sådan ser det ud på skærmen:



Skriv ned

Når man har trykket én gang på **CAPS LOCK**, kan man skrive store bogstaver.

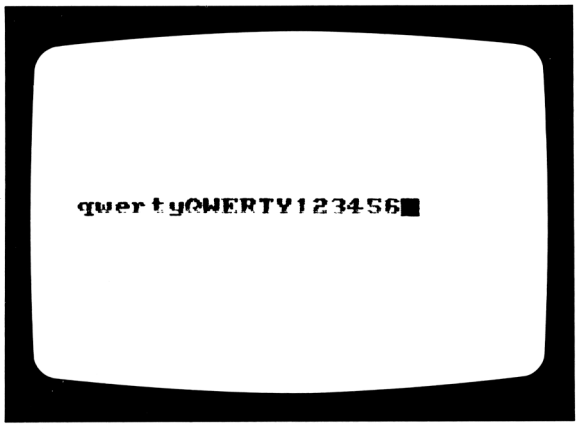
CAPS LOCK

SHIFT Disse to taster er grønne. Hvis du holder en af dem nede, skriver bogstavtasterne store bogstaver og symboltasterne de øverste symboler. Når du slipper **SHIFT** igen, får du de små bogstaver og de nederste symboler eller tal tilbage.

Prøv selv

Tryk på tasterne **123456** én efter én i den øverste række på tastaturet.

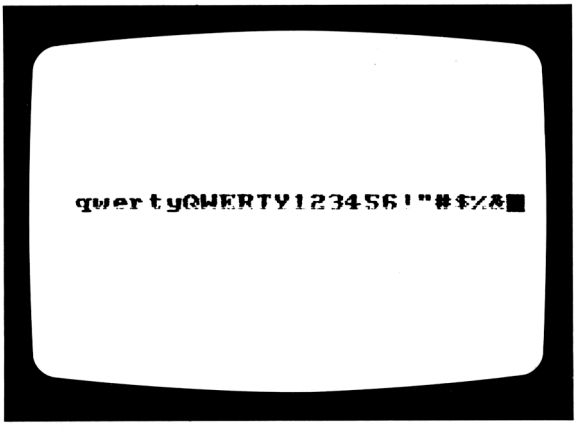
Sådan ser det ud på skærmen:



Prøv selv

Prøv nu at holde **SHIFT** nede, mens du trykker på de samme taster igen: **123456.**

Sådan ser det ud på skærmen:



Skriv ned

Når man holder **SHIFT** nede, kan man skrive store bogstaver og de øverste symboler.

DEL er farvet grøn. Den sletter det tegn, der står til venstre for markøren. Markøren er den lille sorte firkant, som viser, hvor det næste tegn, du skriver, kommer på skærmen.

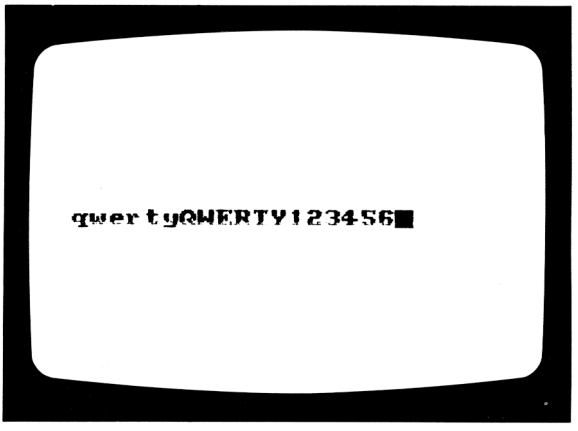
Prøv selv

Tryk på **DEL** tasten seks gange.

Skriv ned

Hver gang man trykker på **DEL**, flytter markøren sig en plads til venstre og sletter det bogstav eller tal, der står på den plads.

Sådan ser det ud på skærmen:

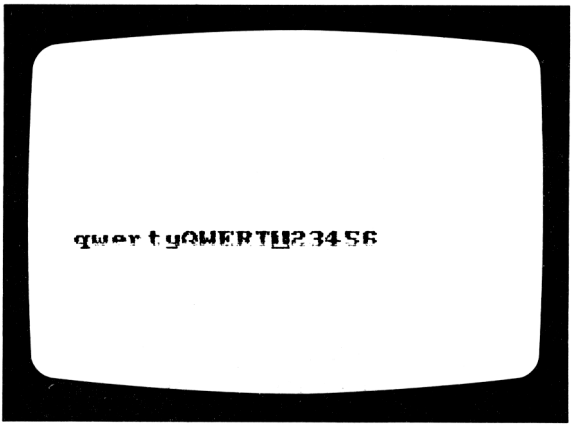


CLR er grøn. Den fjerner det tegn, markøren dækker. Du kan flytte markøren til venstre og højre ved hjælp af tasterne med pile på, der sidder på hver side af **COPY** tasten. Når markøren dækker det tegn, du vil fjerne, trykker du på **CLR,** og tegnet forsvinder. De andre tegn, der står til højre for markøren på den samme linje, bliver rykket således, at den tomme plads fyldes ud.

Prøv selv

Få markøren til at dække bogstavet **Y** ved at trykke 7 gange på tasten med venstrepilen. Tryk så på **CLR.**

Sådan ser det ud på skærmen:

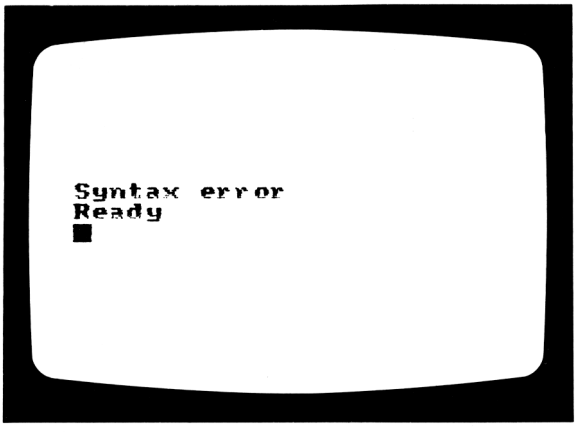


ENTER Der er to af disse taster – en stor og en lille – der sidder mellem taltasterne til højre. De er begge blå. **ENTER** får markøren til at flytte sig til begyndelsen af den næste linje.

Prøv selv

Tryk på **ENTER** (det er ligegyldigt, hvilken af de to taster).

Sådan ser det ud på skærmen:

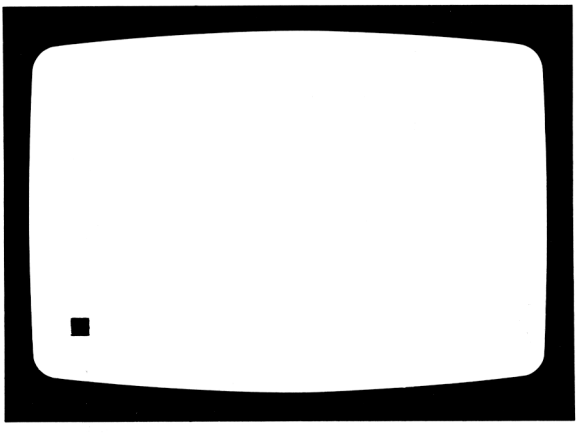


Du skal ikke bekymre dig om beskeden **Syntax error** – vi skal snart lære, hvordan man undgår, at den kommer frem på skærmen.

Prøv selv

Tryk 24 gange på **ENTER**.

Sådan ser det ud på skærmen:



Skriv ned

Når man trykker på **ENTER**, flytter markøren sig en linje ned, og samtidigt "ruller" linjerne opad på skærmen.

Når markøren har nået det nederste af skærmen, skal du lægge mærke til, at ordene oven over den hopper en linje op for hver gang, du trykker på **ENTER** tasten. Det kalder man at "rulle" skærmbilledet.

CTRL er grøn. Når man trykker på den samtidig med andre taster, kan den udføre nogle særlige ting. Her er to af dem:

1. Hold **CTRL** nede og tryk en enkelt gang på **CAPS LOCK.** Hvis du nu trykker på bogstavtasterne får du store bogstaver, mens symboltasterne skriver de øverste symboler. Det virker altså på samme måde som **SHIFT** tasten, men du slipper for at holde **SHIFT** nede hele tiden. Hvis du trykker på både **CTRL** og **CAPS LOCK** endnu en gang, vil datamaskinen vende tilbage til den tilstand, den var i før – den vil lave små bogstaver og tal, hvis **CAPS LOCK** var slået fra og store bogstaver og tal, hvis den var trykket ned.

Skriv ned

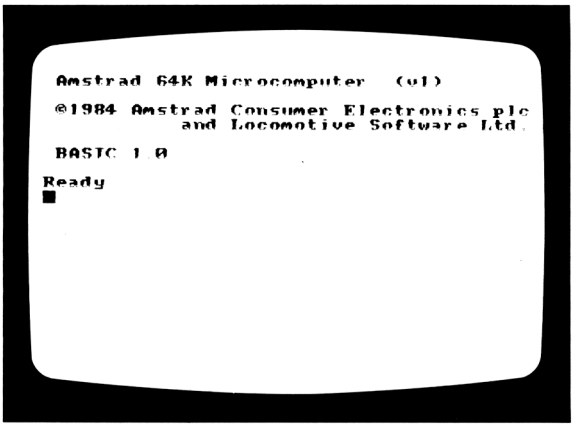
CTRL og **CAPS LOCK** tilsammen har samme virkning som at holde **SHIFT** nede.

2. Hold **CTRL** nede, tryk på **SHIFT** og hold også den nede og tryk så til sidst på **ESC.** Så kommer "vågn op" beskeden på skærmen, ligesom når du tænder for maskinen.

Prøv selv

Tryk samtidig på **CTRL**, **SHIFT** og **ESC.**

Sådan ser det ud på skærmen:



Skriv ned

Når man samtidigt trykker på **CTRL**, **SHIFT** og **ESC** starter maskinen helt forfra. Det virker altså på samme måde som at tænde og slukke for kontakten.

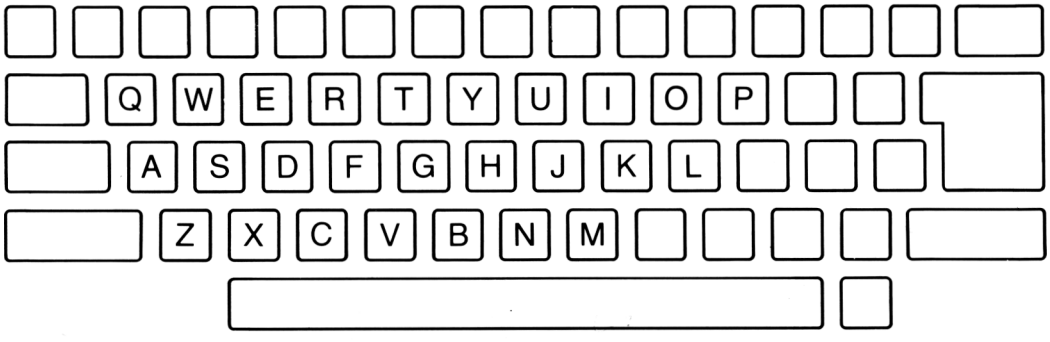
Når man starter maskinen forfra, kaldes det "at sætte den igen". Det får den til at glemme alt, hvad du har fortalt den.

Bogstavtaster

Der er 26 taster, som udskriver bogstaverne i det engelske alfabet:
A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z
og
a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z.
Bagerst i bogen under overskriften: indtastning af danske karakterer, er der et program, der giver de danske bogstaver æ, ø og å. Programmet kan du taste ind, gemme på kassettebånd og herefter bruge, når du vil kunne skrive de danske bogstaver.

Forneden på tastaturet er der også en lang tangent, hvorpå der ikke står noget. Det er mellemrumstangenten, der får markøren til at flytte sig en plads, uden at noget bliver skrevet.

Disse taster sidder på samme måde på tastaturet, som du kender det fra en skrivemaskine.



Prøv selv

Skriv navnene på medlemmerne af din familie og deres fødselsdage. Tryk på **ENTER** i slutningen af hver linje.

Prøv selv

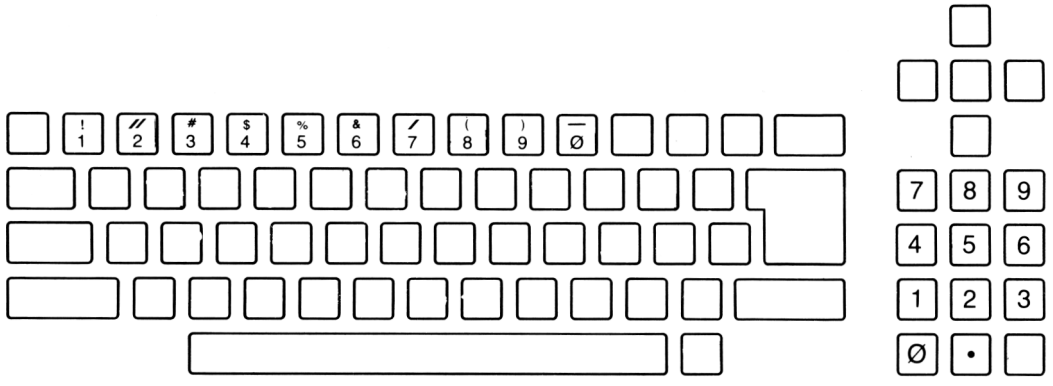
Skriv:
Jeg kan bruge tastaturet.
Tryk på **ENTER.**

Hvis du ikke bryder dig om, at maskinen skriver **Syntax error,** kan du sætte et tal i begyndelsen af hver linje.

Når du er parat til at gå videre med næste afsnit af bogen, tryk da samtidig på **CTRL, SHIFT** og **ESC** for at gøre datamaskinen klar til at starte forfra.

Taltaster

I den øverste række på tastaturet sidder der ti taster, som skriver tallene 1, 2, 3, 4, 5, 6, 7, 8, 9, Ø. Til højre for det store tastatur sidder de ti tal igen sammen med den lille **ENTER** tast og et punktum (som man bruger som decimalkomma). Disse taster tilsammen kaldes for det "numeriske tastatur".



Ø tasten bruges i stedet for nul, så man ikke forveksler det med bogstavet O. Det numeriske tastatur er praktisk, når man skal taste en masse tal ind. Det vil også stadig udskrive tal, når taltasterne på det store tastatur udskriver symboler, fordi du har trykket på **CTRL** og **CAPS LOCK.**

Prøv selv

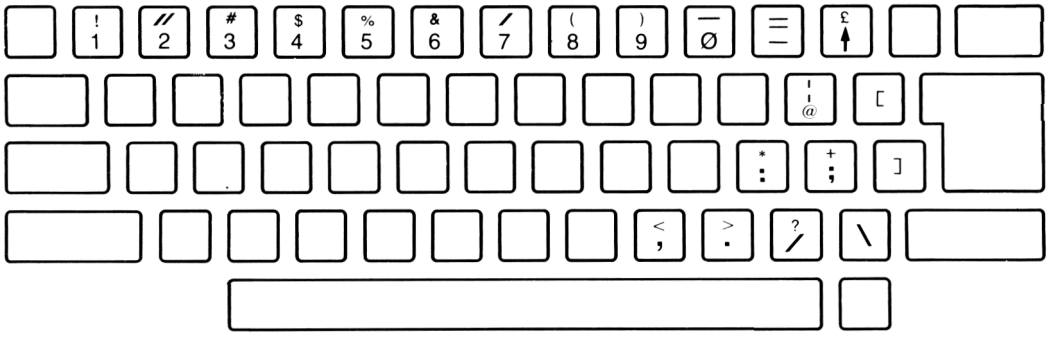
Skriv disse tal: **1**
Tryk på **ENTER** **121**
for enden af **1331**
hver linje. **14641**

Hvis du ikke bryder dig om, at maskinen skriver **Line does not exist,** kan du sætte et bogstav i slutningen af hver linje, før du trykker på **ENTER.**

Når du er parat til at gå videre med næste afsnit, gør da maskinen klar til at starte forfra ved samtidig at trykke på **CTRL, SHIFT** og **ESC.**

Symboltaster

De 11 taster, som skriver symboler, er samlet til højre på det store tastatur. Der er også nogle flere symboler oven over tallene på det store tastatur. For at kunne bruge dem skal du holde en af **SHIFT** tasterne nede eller trykke på **CTRL** og **CAPS LOCK.**



Prøv selv

Skriv det følgende.
Tryk på **ENTER** for enden af hver linje.

1 !
2 "
3 #
4 $
5 %
6 &
7 '
8 (
9)
Ø _

Prøv selv

Skriv det følgende og tryk på **ENTER** for enden af hver linje.

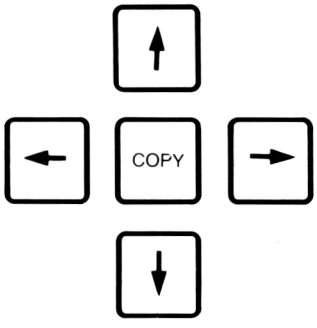
1 Tegnsætning ! " ' ; : , . ?
2 Regnetegn % ( ) + − * / ↑
3 Andre $ & @ £ [ ] \ : _

Når du er parat til at gå videre, gør da maskinen klar til at starte forfra ved samtidig at trykke på **CTRL, SHIFT** og **ESC.**

Editeringstaster

Det er de fire taster med pile på og **COPY** tasten, der sidder samlet oven over det numeriske tastatur.

Amstrad Bruger-vejledning s. G2.8.

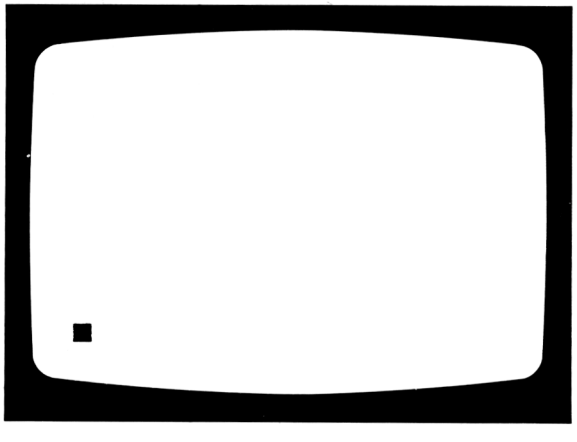


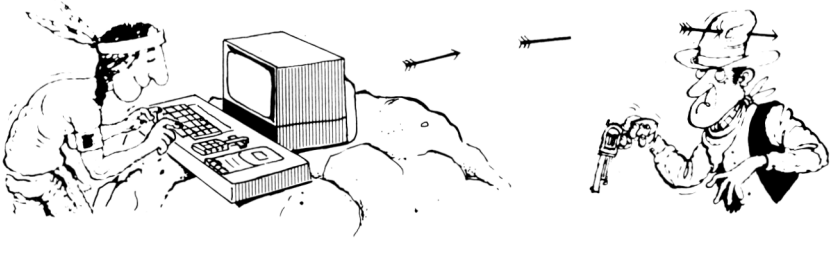
Hver gang man trykker på tasterne med pile på, får man markøren til at flytte sig en plads i den retning, pilen peger. Hvis du bliver ved med at holde tasterne med pile på nede, kan du bevæge dig rundt på hele skærmen. Hvis du holder tasten med opad-pil eller nedad-pil nede længe nok, vil alt det, der stod på skærmen, rulle ud af syne.

Prøv selv

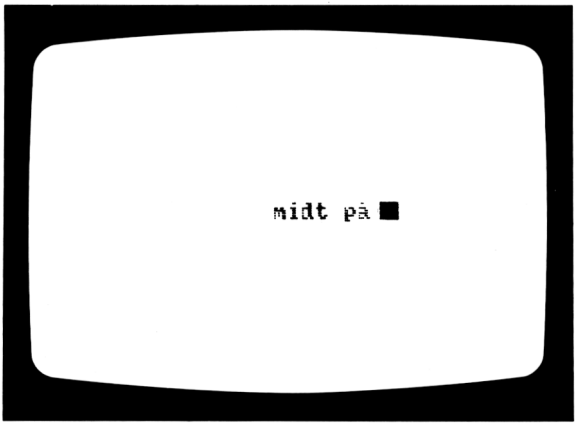
Bliv ved med at trykke på tasten med en nedadrettet pil, indtil alt, hvad du har skrevet, er forsvundet fra skærmen.

Sådan ser det ud på skærmen:





Sådan ser det ud på skærmen:



Prøv selv

Brug tasterne med pile på til at flytte markøren til midten af skærmen. Skriv nu: **midt på**.

Nu ved du, hvordan man kan skrive overalt på skærmen.

Hvis du holder **SHIFT** nede, mens du trykker på en tast med pil på, kommer der en anden markør frem, som kan bevæge sig i den retning, pilen peger. Dette er kopi-markøren.

Sådan ser det ud på skærmen:

Prøv selv

Gør maskinen klar til at starte forfra. Hold **SHIFT** nede og tryk 8 gange på opad-pilen.

Når alle taster er fundet
-er snart det hele vundet.

Prøv selv

Tryk på **COPY** tasten indtil kopi-markøren står i slutningen af den øverste linje.

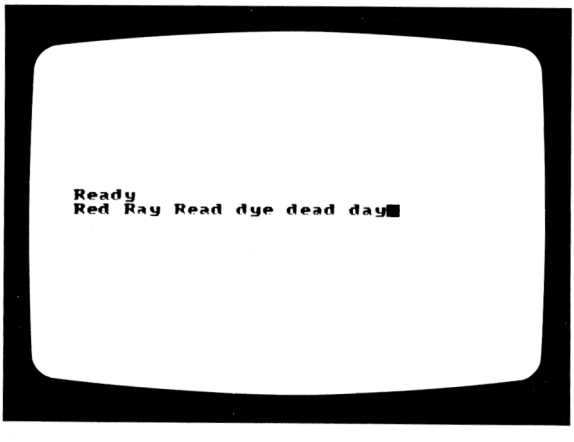
Sådan ser det ud på skærmen:

Læg mærke til, hvordan datamaskinen kopierer et bogstav, et tal eller et mellemrum, hver gang du trykker på **COPY** tasten. Du kan for øvrigt godt slippe **SHIFT**, mens du bruger **COPY**!

Prøv selv

Tryk på **ENTER**.
Brug nu **SHIFT**, pilene og **COPY** tasten samt mellemrumstangenten til at kopiere følgende ord fra "ready" tegnet:
Red Ray Read dye dead day

Sådan ser det ud på skærmen:



NÅR DU SKAL RIGTIGT I GANG

S. T. RØMER

Mine råd

Når du begynder at få maskinen til at udføre ting for dig, skal du altid kontrollere dem meget omhyggeligt. Datamaskinen er meget præcis, når den skal forstå noget. Hvis der mangler et mellemrum, eller et punktum står forkert, eller måske et bogstav er forkert, betyder det, at maskinen ikke forstår dig.

Når det sker, så *undersøg* det. Opspor alle muligheder for fejl, indtil du har det hele rigtigt. I begyndelsen tager det sikkert nogen tid at opspore fejlene, men efterhånden bliver du ikke kun hurtigere til det – du laver også færre fejl.

Skriv de ting ned, du synes, er gode at huske. Så kan du let finde dem, når du har brug for det. Endelig er der to ting, som jeg synes, er meget vigtige at huske.

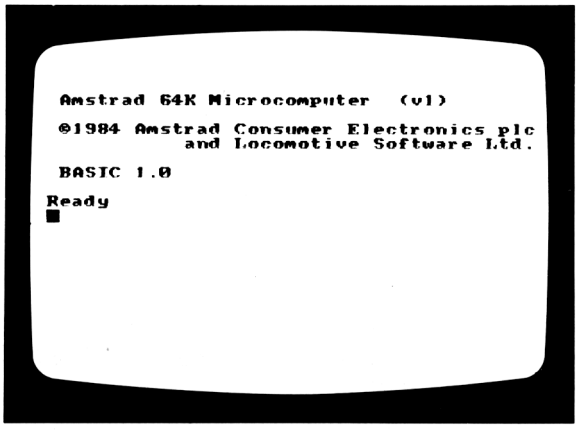
1. Husk at datamaskinen er meget nøjeregnende med, hvor man sætter mellemrum.
2. Husk at trykke på **ENTER** efter hver ordre.

Starkodder Thorbjørn Rømer

I dette afsnit skal vi finde ud af at anvende datamaskinen på fire forskellige måder:

1. Datamaskinen som skrivemaskine.
Vi skal udforske, hvordan man skriver tal og tegn på skærmen, og hvordan man på forskellige måder kan skille dem fra hinanden.
2. Datamaskinen som regnemaskine.
Vi skal lære, hvordan man lægger sammen, trækker fra, ganger og dividerer tal ved hjælp af maskinen.
3. Datamaskinen til lagring af information.
Vi skal finde ud af, hvordan man lægger tal ind i maskinens hukommelse.
4. Datamaskinen til søgning af information.
Vi skal lære, hvordan man får maskinen til at spørge efter information, som den derpå lægger ind i hukommelsen.

Når du tænder for Amstrad, står der dette på skærmen:



Amstrad Microcomputer (v1) – datamaskinens mærke og hvilken version, den er.

64K – størrelsen af maskinens hukommelse.

© 1984 Amstrad Consumer Electronics plc and Locomotive Software Ltd. – copyright linjen. Den betyder, at du ikke må lave kopier af maskinen eller de programmer, du har fået med til den uden at få lov af Amstrad og Locomotive Software.

BASIC 1.Ø – det programmeringssprog, datamaskinen anvender.

Ready – det signal, der fortæller dig, at maskinen er klar til at modtage ordrer.

■ Dette er markøren. Den viser dig, hvor de bogstaver, du skriver, kommer på skærmen.

Datamaskinen som skrivemaskine

For at få maskinen til at udskrive noget, bruger man **print.** Når du har skrevet, hvad du vil, husk da at trykke på **ENTER.**

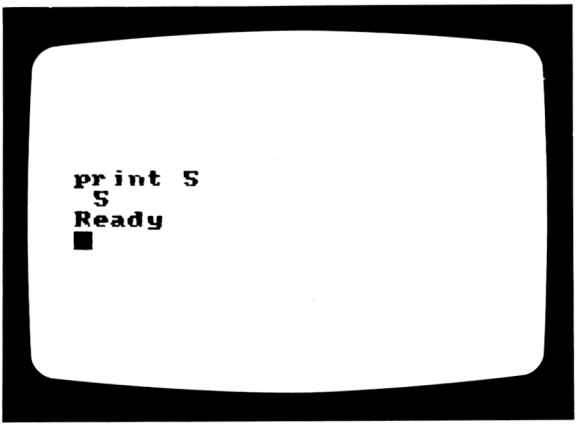
Amstrad Bruger-vejledning kap. 8, s. 54.

Udskrivning af tal

Prøv selv

Skriv **print 5** og tryk derefter på **ENTER.** Husk at lave et mellemrum efter **print.**

Sådan ser det ud på skærmen:



Skriv ned

Når man vil have et tal udskrevet, skal man skrive **print** efterfulgt af et mellemrum, derefter tallet og til sidst trykke på **ENTER.**

Prøv selv

Skriv **print 675** og tryk på **ENTER.**

Prøv selv

Skriv på samme måde nogle andre tal, du selv finder på.

Sådan ser det ud på skærmen:

```
print 675
 675
Ready
```

Udskrivning af symboler, bogstaver og ord

Prøv selv

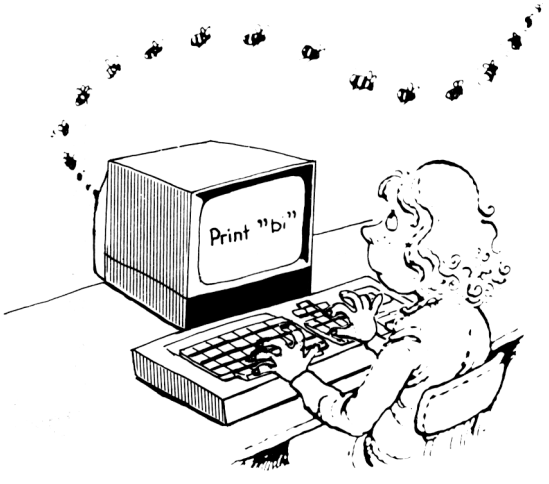
Skriv **print "?"** og tryk på **ENTER.**

Sådan ser det ud på skærmen:

```
print "?"
?
Ready
```

Skriv ned

For at få udskrevet et symbol, et bogstav eller et ord skriver man **print** efterfulgt af et mellemrum, herefter symbolet, bogstavet eller ordet i citationstegn og trykker til sidst på **ENTER.**



Prøv selv

Skriv **print "bi"** og tryk på **ENTER.**

Sådan ser det ud på skærmen:

```
print "bi"
bi
Ready
■
```

Prøv selv

Brug **print** til at få maskinen til at udskrive dit navn.

Mellemrum mellem tal

Prøv selv

Skriv **print 1,2,3** og tryk på **ENTER.**

Sådan ser det ud på skærmen:

```
print 1,2,3
 1            2            3
Ready
■
```

Prøv selv

Skriv **print 1;2;3** og tryk på **ENTER.**

Sådan ser det ud på skærmen:

```
print 1;2;3
 1  2  3
Ready
■
```

Prøv selv

Skriv **print 12,,,34** og tryk på **ENTER.**

Prøv selv

Prøv på de samme måder at lave mellemrum mellem nogle tal, du selv finder på.

Sådan ser det ud på skærmen:

```
print 12,,,34
 12
 34
Ready
■
```

Skriv ned

Symbolet:
, udskriver tallene med store mellemrum mellem
; udskriver tallene adskilt med to mellemrum
,,, udskriver på to linjer

Gå ikke helt i stå
-UNDERSØG TINGENE.

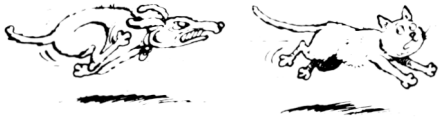
Mellemrum mellem ord

Prøv selv

Skriv **print "hund","kat"** og tryk på **ENTER.**

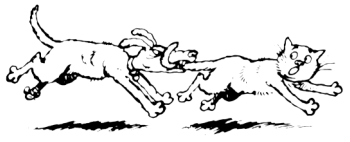
Sådan ser det ud på skærmen:

```
print "hund","kat"
hund          kat
Ready
■
```

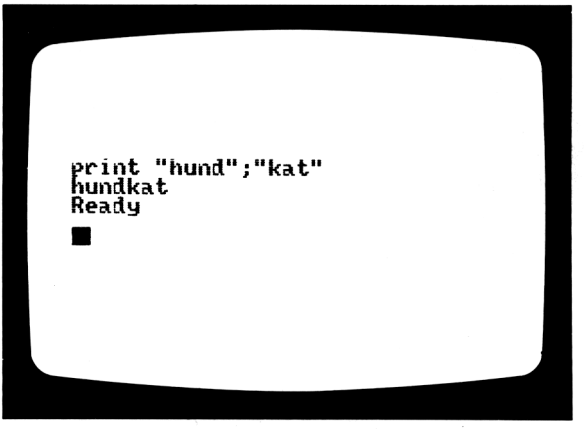


Prøv selv

Skriv **print "hund";"kat"** og tryk på **ENTER.**



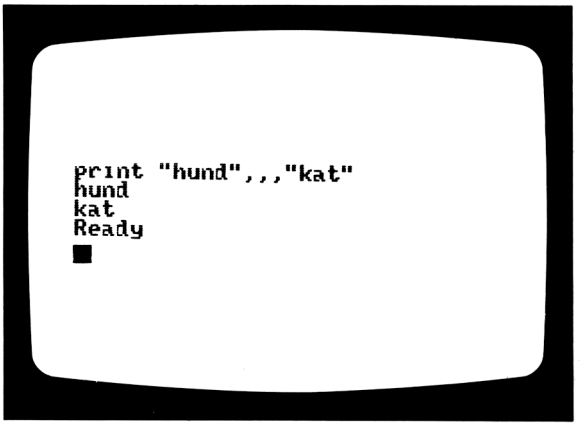
Sådan ser det ud på skærmen:



Prøv selv

Skriv **print "hund",,,"kat"** og tryk på **ENTER.**

Sådan ser det ud på skærmen:



Prøv selv

Forsøg på samme måde at lave forskellige mellemrum mellem dit fornavn og efternavn.

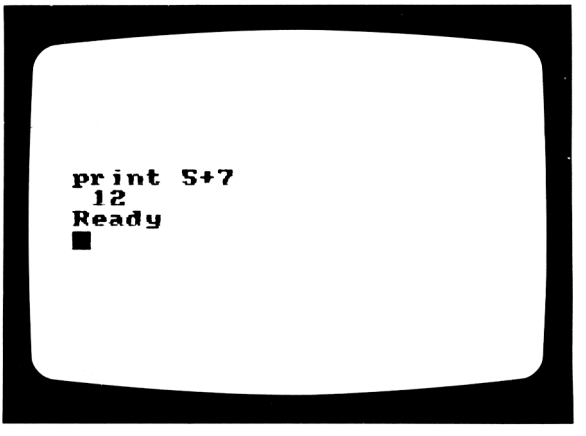
Skriv ned

Symbolet:
, udskriver ord med store mellemrum mellem
; udskriver ord uden mellemrum mellem
,,, udskriver ord på to linjer

Datamaskinen som regnemaskine

print ordren kan også få maskinen til at fungere som en regnemaskine. Man kan stille den et spørgsmål, hvorefter den giver svaret. Husk at trykke på **ENTER,** når du har skrevet det, du vil.

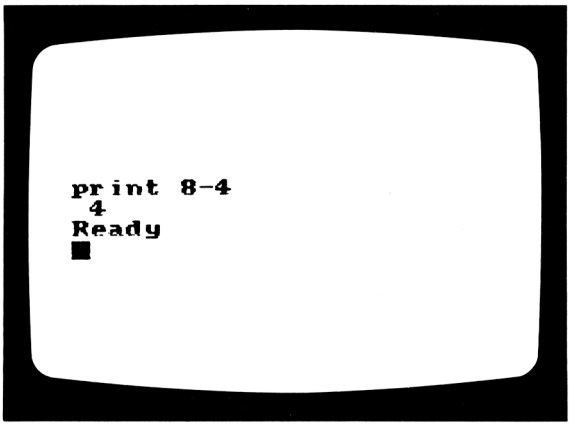
Sådan ser det ud på skærmen:



Prøv selv

Skriv **print 5+7** og tryk på **ENTER.**

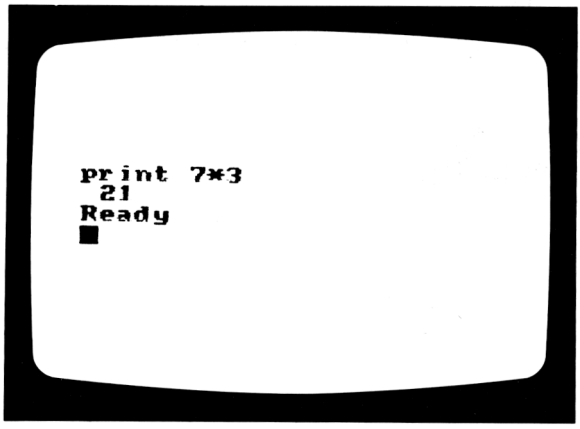
Sådan ser det ud på skærmen:



Prøv selv

Skriv **print 8–4** og tryk på **ENTER.**

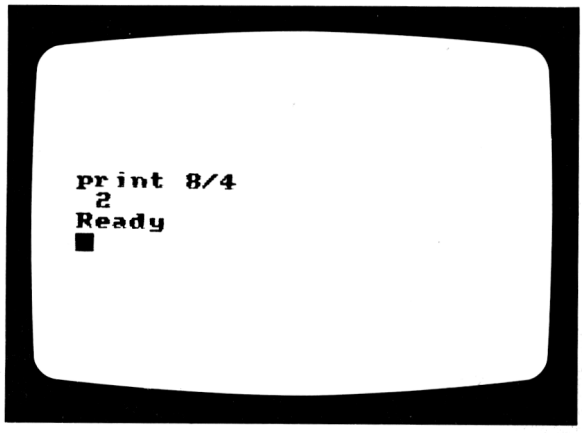
Sådan ser det ud på skærmen:



Prøv selv

Skriv **print 7*3** og tryk på **ENTER.**

Sådan ser det ud på skærmen:



Prøv selv

Skriv **print 8/4** og tryk på **ENTER.**

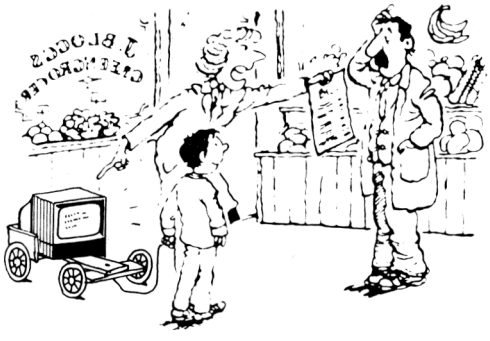
Prøv selv

1. Brug de samme ideer til selv at lave nogle regnestykker.
2. Brug datamaskinen til at regne en kassebon fra et supermarked efter.

Skriv ned

De symboler, man bruger, er:

\+ lægge sammen (addere)
\– trække fra (subtrahere)
* gange (multiplicere)
/ dele (dividere)



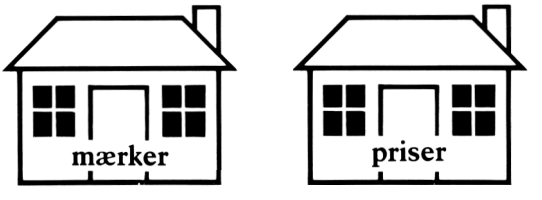
Datamaskinens hukommelse

Du kan også bruge datamaskinen til at huske information. Det er vigtigt at vide nøjagtigt, hvordan maskinen gør dette, før du går videre med næste del af afsnittet.

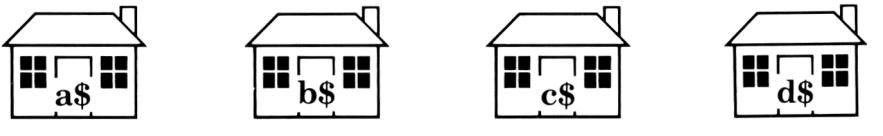
Ligesom du bor på en adresse, din ven på en anden og din lærer på en helt tredje, således bruger datamaskinen også et system af adresser til at lagre information.

Adresserne i dens hukommelse kan enten være bogstaver eller ord. For eksempel kan den lagre tal på de følgende adresser:

eller



Den lagrer ord på adresser som disse:



Skriv ned

Datamaskinen bruger **a**, **b**, **c**, **d** o.s.v. som adresser til at lagre tal og **a$**, **b$**, **c$**, **d$** o.s.v. som adresser til at lagre ord.

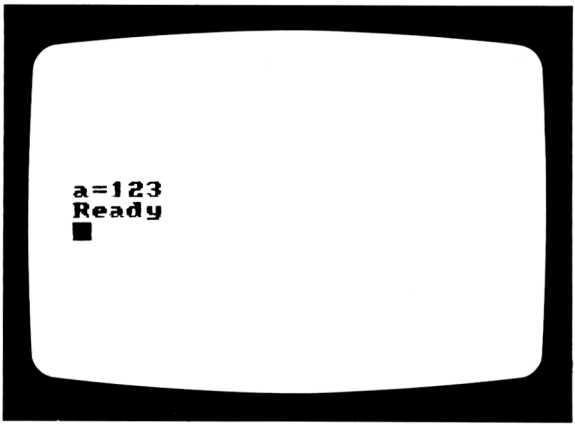
Datamaskinen til lagring af information

= tegnet bruges til at lægge tal eller ord ind i datamaskinens hukommelse.

Prøv selv

Skriv **a=123** og tryk på **ENTER.**

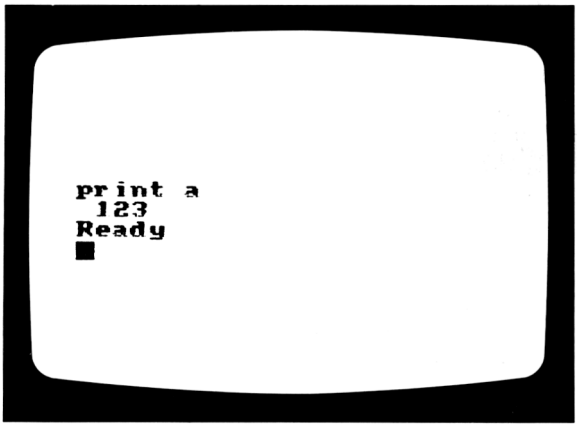
Sådan ser det ud på skærmen:



Prøv selv

Skriv **print a** og tryk på **ENTER.**

Sådan ser det ud på skærmen:



a=123 lægger tallet 123 ind i datamaskinens hukommelse på adressen **a. print a** udskriver det tal, som ligger på adressen **a** i maskinens hukommelse.

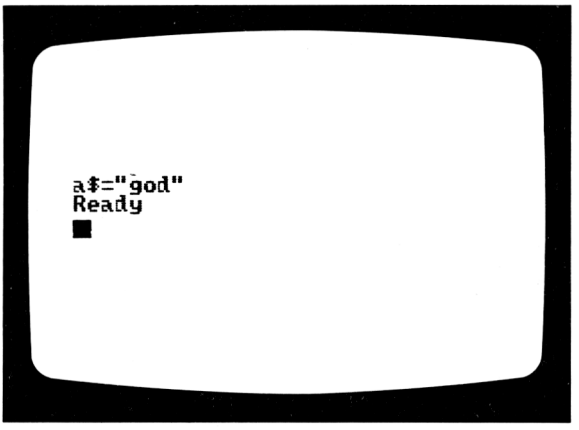
Prøv selv

Brug = tegnet til at lægge tallet 12 ind på adresse **b** og tallet 4 ind på adresse **c.** Skriv så **print b+c,,,b–c,,,b*c,,,b/c** og tryk på **ENTER.**

Prøv selv

Skriv **a$="god"** og tryk på **ENTER.**

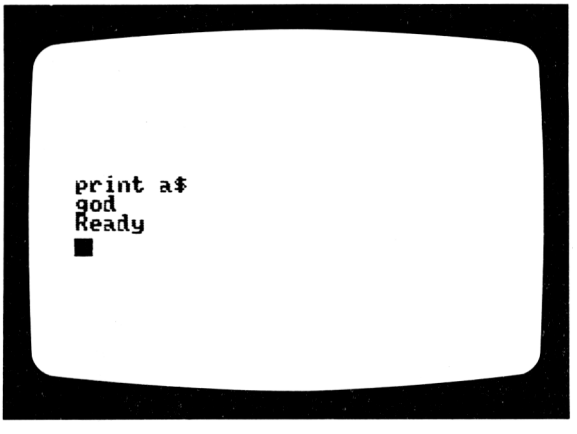
Sådan ser det ud på skærmen:



Prøv selv

Skriv **print a$** og tryk på ENTER.

Sådan ser det ud på skærmen:



a$="god" lægger ordet **god** ind i datamaskinens hukommelse på adressen **a$. print a$** udskriver det ord, der ligger på adressen **a$** i maskinens hukommelse.

Prøv selv

Brug = tegnet til at lægge **jeg** ind på adressen **b$** og **er** ind på adressen **c$.**
Skriv **print c$,b$,a$** og tryk på **ENTER.**
Skriv **print b$,c$,a$** og tryk på **ENTER.**

Datamaskinen til søgning af information

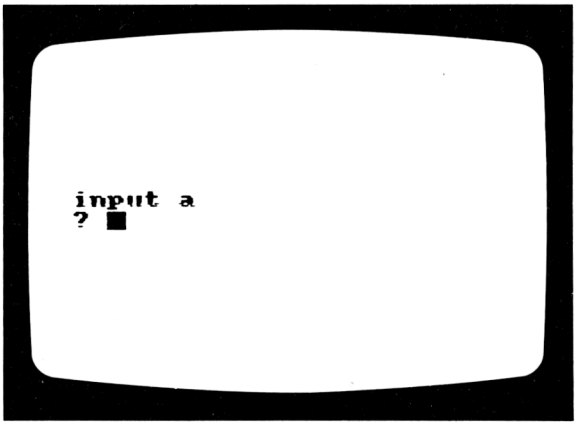
Amstrad Bruger-vejledning G2.6.

Input ordren bruges til at få maskinen til at spørge efter tal eller ord, som den derpå lægger ind på en adresse i hukommelsen.

Prøv selv

Skriv **input a** og tryk på **ENTER.**

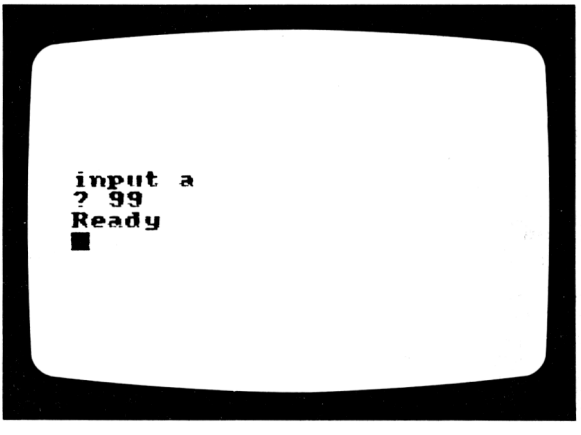
Sådan ser det ud på skærmen:



Prøv selv

Skriv **99** og tryk på **ENTER.**

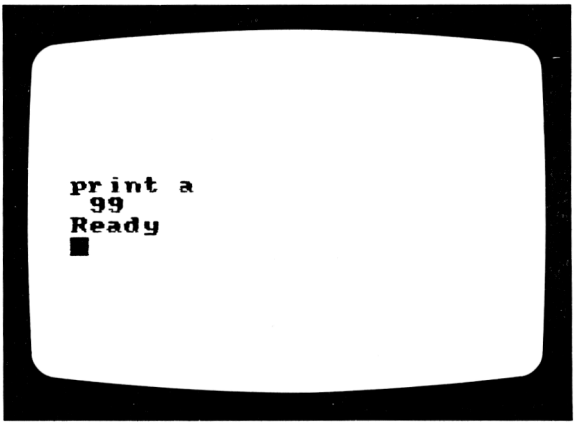
Sådan ser det ud på skærmen:



Prøv selv

Skriv **print a** og tryk på **ENTER.**

Sådan ser det ud på skærmen:



input a fik maskinen til at spørge efter et tal. Når man har skrevet tallet **99,** lægger maskinen **99** ind på adressen **a** i hukommelsen. **print a** udskriver det tal, som ligger på adressen **a.**

Prøv selv

Brug **input** ordren til at få maskinen til at spørge efter nogle tal og lægge dem ind i hukommelsen på adresserne **b** og **c.** (**?** betyder, at der spørges efter et tal). Skriv **print b,c,b+c,b*c** og tryk på **ENTER.**

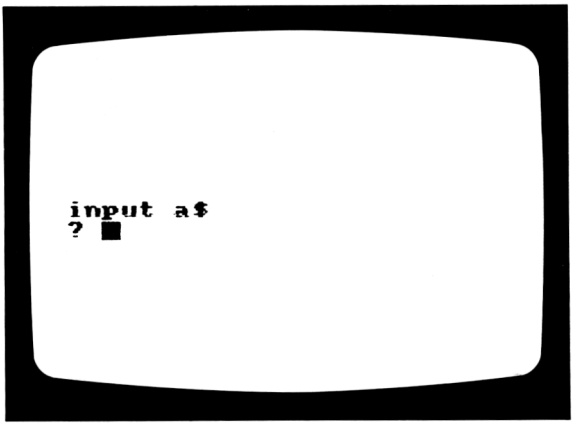
Skriv ned

1. **input a** får maskinen til at spørge efter et tal.
2. **?** tegnet betyder, at der spørges efter tallet.
3. Det tal, man giver maskinen, lægges ind i hukommelsen på adressen **a.**

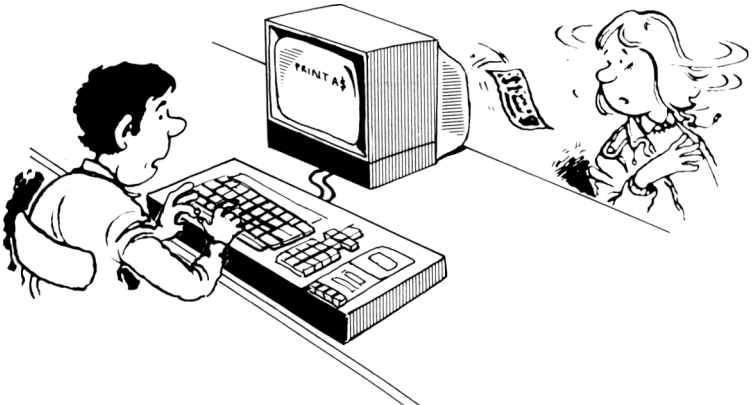
Prøv selv

Skriv **input a$** og tryk på **ENTER.**

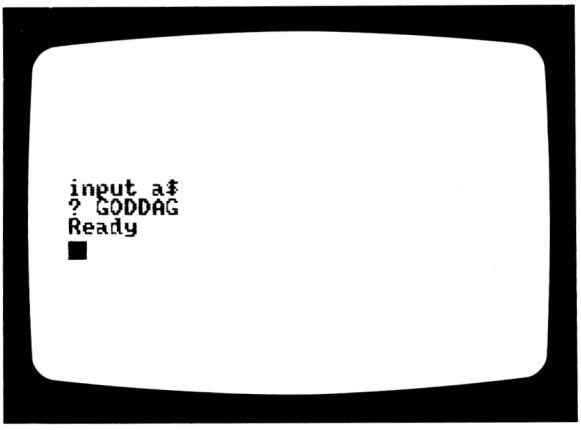
Sådan ser det ud på skærmen:



Selv om maskinen ikke er nem, giv ikke op, men prøv dig frem.



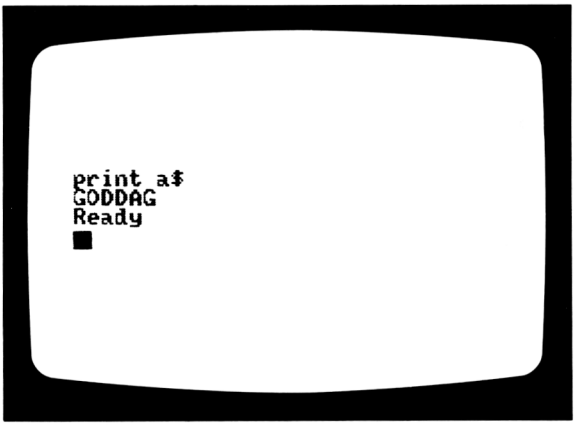
Sådan ser det ud på skærmen:



Prøv selv

Skriv **GODDAG** og tryk på **ENTER.**

Sådan ser det ud på skærmen:



Prøv selv

Skriv **print a$** og tryk på **ENTER.**

Du kan få maskinen til at gøre lige, hvad du vil!

input a$ fik maskinen til at spørge efter et ord. Når man taster ordet **GODDAG** ind, lægger maskinen **GODDAG** ind i hukommelsen på adressen **a$. print a$** udskriver det ord, der ligger i hukommelsen på adressen **a$.**

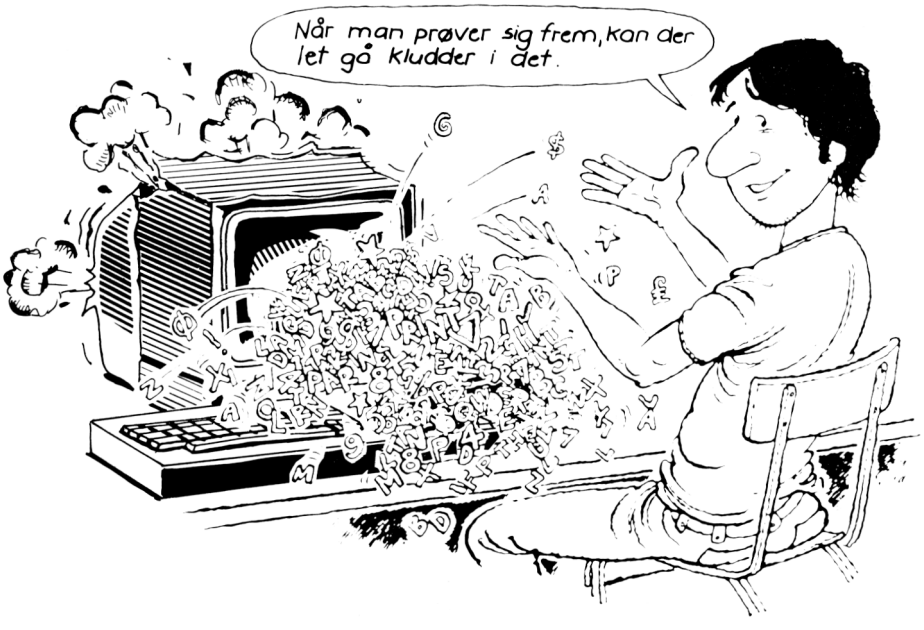
Prøv selv

Brug **input** ordren til at lægge dit navn ind i hukommelsen på adresse **b$.** (**?** betyder, at der spørges efter et ord). Skriv **print a$,b$** og tryk på **ENTER.**

Skriv ned

1. **input a$** får maskinen til at spørge efter et ord.
2. **?** tegnet er den måde, hvorpå maskinen spørger efter et ord.
3. Det indtastede ord lægges på adressen **a$** i hukommelsen.

SÆRLIGE MULIGHEDER



FREDE ARVETRIP

Mine råd

Et af mine råd er, at du skal kigge opmærksomt på de nye ordrer. Når du synes, du forstår ideen i dem, skal du afprøve dem. Du lærer, hvad der ser rigtigt ud. Når man prøver sig frem, kan der let gå kludder i det. Men lad dig ikke slå ud: prøv igen. Tænk på, når det lykkes for dig, og glem, når det mislykkes.
Du skal afprøve alle de ideer, du får. Vi skal udforske linjer, vinkler, figurer, farver – ja, oven i købet musik. Du kan tegne næsten alt på skærmen og farve det på forskellige måder.
Og for at fejre dit mesterværk kan du komponere en melodi eller endda en hel symfoni.
Når du får en god idé, skal du huske at skrive den ned, for at du senere kan bruge den igen.

Frede Arvetrip

Skift af skærmtilstand

Tegnene på skærmen bliver udskrevet på bestemte måder, der afhænger af, hvilken tilstand maskinen arbejder i. Når du tænder for din Amstrad, er den i tilstand 1. I tilstand 1 er der plads til 4Ø tegn på en linje og 25 linjer på hele skærmen. Men datamaskinen kan lige så godt arbejde i tilstandene Ø og 2.

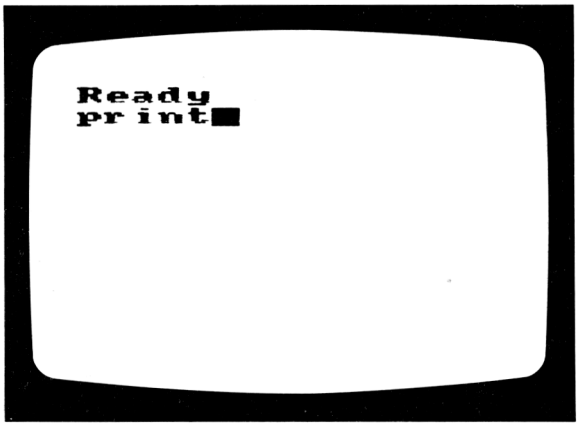
For at skifte tilstand, skal man skrive **mode** (tilstand på engelsk) efterfulgt af et mellemrum og herefter nummeret på den tilstand, du vil have. Hvis du for eksempel skriver **mode Ø** og trykker på **ENTER,** får du maskinen i tilstand Ø. I denne tilstand er der kun plads til 2Ø tegn på hver linje. I tilstand 2 er der plads til 8Ø tegn på hver linje. Ligegyldigt hvilken tilstand, du vælger, er der plads til 25 linjer på hele skærmen.

Amstrad Bruger-vejledning kap.5, s.3.

Prøv selv

Skriv **mode Ø** og tryk på **ENTER.**
Skriv nu **print** (læg mærke til forskellen).

Sådan ser det ud på skærmen:

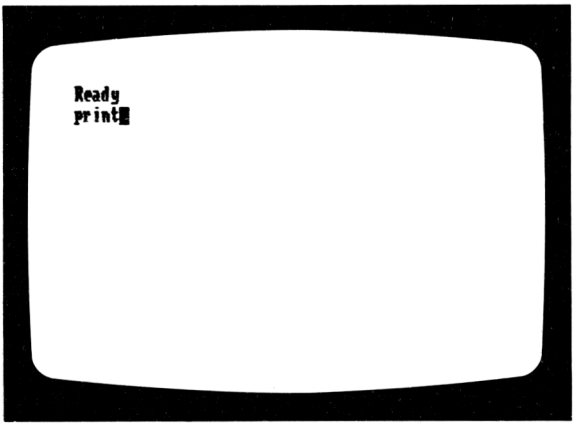


Hvis du den rette tilstand får - du længst ad vejen når.

Prøv selv

Skriv **mode 2** og tryk på **ENTER.**
Skriv **print** igen.

Sådan ser det ud på skærmen:



Tegning på skærmen

Amstrad Bruger-vejledning s. G3.10.

draw ordren bruges til at tegne linjer og figurer. Når man vil anvende datamaskinen til at tegne, skal **draw** ordren efterfølges af et mellemrum, hvorefter man skriver to tal adskilt med et komma. For eksempel:

draw 32Ø,2ØØ
draw 639,Ø

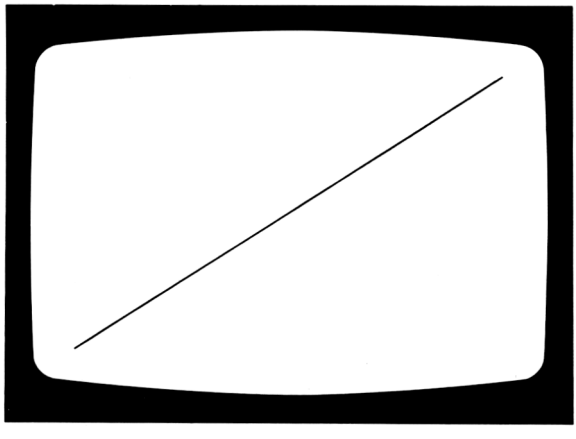
Hvis vi forestiller os, at det, du tegner, skal afbildes på skærmen, kan det første tal ligge mellem Ø og 639. Det andet tal skal ligge mellem Ø og 399.

draw ordren virker på samme måde som, når man tegner uden at løfte blyanten fra papiret. Det første tal efter **draw** ordren står for afstanden vandret på skærmen, mens det andet tal står for den lodrette afstand begyndende i det nederste venstre hjørne.

Prøv selv

Skriv **mode 2** og tryk på **ENTER.**
Skriv **draw 639,399** og tryk på **ENTER.**

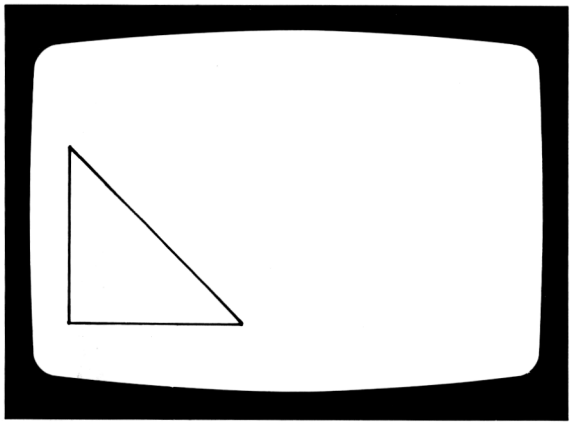
Sådan ser det ud på skærmen:



Prøv selv

Skriv **mode 2** og tryk på **ENTER.**
Skriv **draw 32Ø,Ø** og tryk på **ENTER.**
Skriv **draw Ø,2ØØ** og tryk på **ENTER.**
Skriv **draw Ø,Ø** og tryk på **ENTER.**

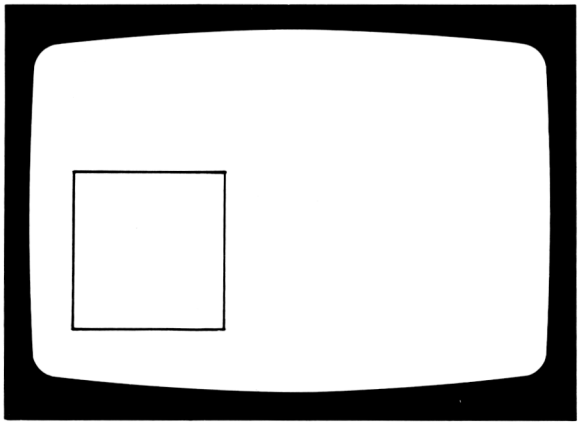
Sådan ser det ud på skærmen:



Prøv selv

Skriv **mode 2** og
tryk på **ENTER.**
Skriv **draw 2ØØ,Ø**
og tryk på **ENTER.**
Skriv **draw 2ØØ,2ØØ**
og tryk på **ENTER.**
Skriv **draw Ø,2ØØ**
og tryk på **ENTER.**
Skriv **draw Ø,Ø** og
tryk på **ENTER.**

Sådan ser det ud på skærmen:



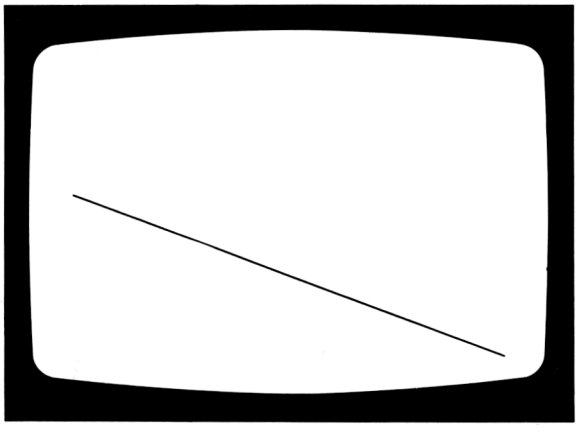
Orientering på skærmen

move ordren bruges til at bevæge sig rundt på skærmen uden at tegne linjer. Når du bruger **draw** ordren, opdager du, at du nogle gange har brug for at kunne bevæge dig til et andet punkt på skærmen uden hermed at tegne en linje. Det kan man gøre ved at bruge **move.** Ligesom **draw**-ordren kunne sammenlignes med at tegne uden at løfte blyanten fra papiret, således gør **move** det muligt for dig at løfte blyanten og tegne videre et andet sted.

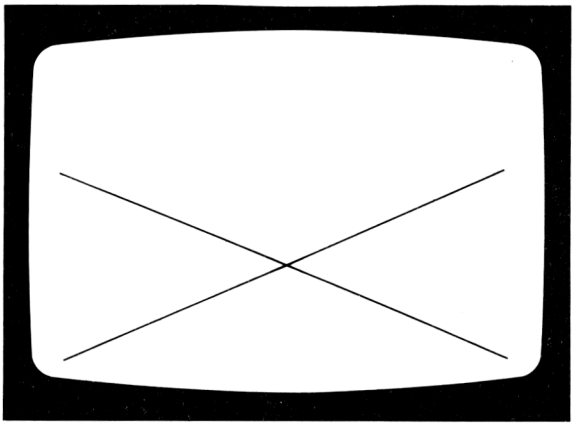
Prøv selv

Skriv **mode 2** og
tryk på **ENTER.**
Skriv **move Ø,2ØØ**
og tryk på **ENTER.**
Skriv **draw 639,Ø**
og tryk på **ENTER.**

Sådan ser det ud på skærmen:



Sådan ser det ud på skærmen:



Prøv selv

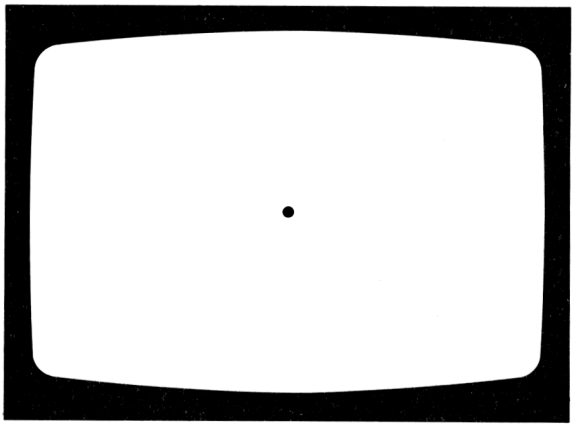
Skriv **mode 2** og
tryk på **ENTER.**
Skriv **draw 639,200**
og tryk på **ENTER.**
Skriv **move 639,0**
og tryk på **ENTER.**
Skriv **draw 0,200**
og tryk på **ENTER.**

Afmærkning af punkter

Amstrad Bruger-vejledning s. G3.9.

plot ordren bruges til at bevæge sig hen til et punkt på skærmen og afmærke det. Ligesom med **move,** tegnes der ikke en linje på vej hen til punktet. **plot** kan bruges til at fremhæve enkelte punkter på skærmen og til at afbilde figurer, der ikke består af lige linjer.

Sådan ser det ud på skærmen:



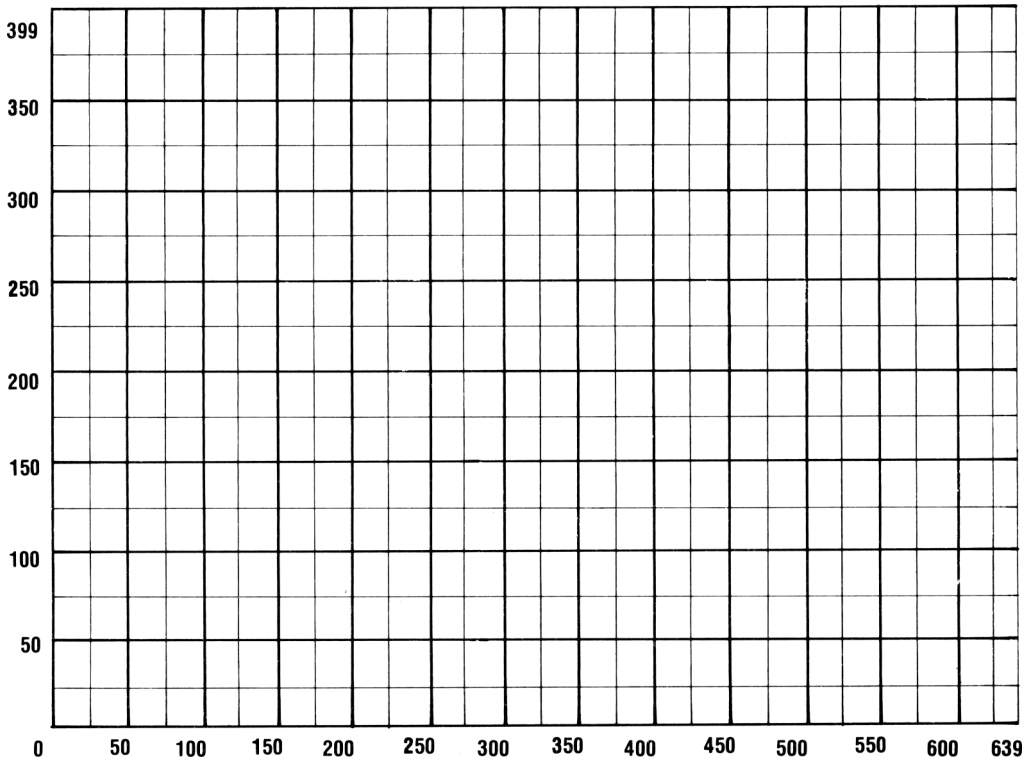
Prøv selv

Skriv **mode 0** og
tryk på **ENTER.**
Skriv **plot 320,200**
og tryk på **ENTER.**
Prøv at afmærke
punkter i tilstandene 1
og 2.

Skriv ned

draw, move og **plot**
skal efterfølges af et
mellemrum og to tal.
Der er 640 punkter
vandret på skærmen og
400 lodret. Der skal
være komma mellem
de to tal.

For at hjælpe dig med selv at fremstille tegninger, er der her en plan over skærmbilledet, der viser de vandrette og lodrette afstande med start i nederste venstre hjørne.



Prøv selv

Brug **draw, move** og **plot** ordrene til at tegne et hus eller et vejskilt.

Hvis du er interesseret i nærmere at udforske metoder som **draw, move** og **plot,** kan du slå op under **drawr, mover, origin** og **window** i Brugervejledningen. Du kan også kigge på afsnittet "Brug af skærmen" i opslagsdelen af denne bog.

Farveskift

Amstrad Bruger-vejledning s. G3.1.

Der er mulighed for at bruge 27 forskellige farver på Amstrad Mikrodatamaten. Du kan selvfølgelig kun se dem, hvis du har en farveskærm eller bruger et farvefjernsyn. Hvis du har en grøn skærm, kan farverne ses som forskellige nuancer af grønt.

Når man tænder for datamaskinen, står tegnene med gult på en mørkeblå baggrund.

Prøv selv

Skriv **border Ø** og tryk på **ENTER.**
Skriv **border 15** og tryk på **ENTER.**

I denne "Prøv selv" øvelse kommer der først en sort kant rundt om skærmbilledet og herefter en orange.

Prøv selv

Skriv **border 3,4** og tryk på **ENTER.**

I denne "Prøv selv" øvelse får du en kant, der blinker skiftende mellem farverne rød og lilla. Når du bliver træt af at kigge på den, kan du skrive **border 1** for at få den faste blå kant tilbage.

Hver farve har et nummer, der kaldes "farvenummeret". Sort har for eksempel farvenummeret Ø og blå har nummeret 1. Du kan finde en liste over alle farvenumrene i *Amstrad Bruger-vejledningen s. G3.2.*

Skriv ned

border efterfulgt af et mellemrum og ét tal skifter kantfarven til en ensfarvet, fast farve.
border efterfulgt af et mellemrum og 2 tal med komma mellem giver en blinkende kant.

Prøv selv

Skriv **paper 2** og tryk på **ENTER**.
Skriv **cls** og tryk på **ENTER**.

I denne "Prøv selv" øvelse skifter du baggrundsfarven inden for kanten til klar blårød. **cls** får maskinen til at slette skærmbilledet.

Skriv ned

cls efterfulgt af **ENTER** visker skærmen ren.

Prøv selv

Skriv **pen 3** og tryk på **ENTER**.
Skriv **cls** og tryk på **ENTER**.

I denne "Prøv selv" skifter den farve, hvormed tegnene skrives, fra gul til rød. Nu bliver alt, hvad du skriver, rødt.

Det afhænger af maskintilstanden, hvor mange farver du på samme tid kan anvende på skærmen. Når man tænder for maskinen, er den i tilstand 1. I tilstand 1 kan du bruge fire farver samtidig, når man ikke tæller kanten med. I tilstand Ø kan du bruge helt op til 16 forskellige farver, mens du i tilstand 2 kun kan anvende 2 (når man stadig ikke tæller kanten med).

Prøv selv

Skift til tilstand Ø og prøv at ændre farverne for **paper** og **pen** ved at bruge farvenumrene mellem Ø og 15. Visk skærmen ren, hver gang du har ændret **paper** nummeret.

Skriv ned

Ordrene **paper** og **pen** bruges til at vælge baggrundsfarve og skrivefarve på skærmen. Hvis den farve, du skriver med, er den samme som baggrundsfarven, kan du ikke se, hvad du skriver.

Prøv selv

Gør maskinen klar til at starte forfra.
Skriv **ink 1,16** og tryk på **ENTER.**

Alt, hvad der er skrevet på skærmen, skifter i denne "Prøv selv" øvelse på én gang farve til lyserød.

Prøv selv

Skriv **ink 2,9** og tryk på **ENTER.**
Skriv **pen 2** og tryk på **ENTER.**

I denne "Prøv selv" øvelse skifter alt, hvad du skriver efter **pen 2** farve til grønt. Det, der stod før, er stadig lyserødt.

Prøv selv

Skriv **ink 3,14** og tryk på **ENTER.**
Skriv **paper 3** og tryk på **ENTER.**
Skriv **cls** og tryk på **ENTER.**

I denne "Prøv selv" øvelse viskes skærmen ren, og baggrundsfarven skifter til lyseblå.

Når du tænder for datamaskinen, kan du i tilstand 1 bruge fire farver for **paper** og **ink,** nemlig: blå, gul, blårød og lyserød. Hvis du skriver **ink** efterfulgt af et mellemrum og 2 tal med komma mellem, kan du i stedet vælge nogle andre farver. Når man bruger **ink** ordren, er det ligesom at skifte farve i kuglepennen og skrive på forskelligt farvet papir.

Skriv ned

Med **ink** ordren kan man ændre den farveskala, som benyttes til **pen** og **paper.**

Komponering af musik

Amstrad Bruger-vejledning s. G3.14.

sound ordren bruges til at spille toner med. Den virker lige godt i alle tre tilstande, så der er ingen grund til at vælge en særlig tilstand. **sound** ordren skal efterfølges af et mellemrum og herpå mindst to tal adskilt med kommaer. For eksempel:

sound 1,5Ø6
sound 1,6Ø,1ØØ
sound 1,239,2ØØ,2

Det første tal vælger kanal. Du behøver kun andre kanaler end nummer 1, hvis du vil spille flere toner på én gang.
Det andet tal vælger, hvor højt eller lavt tonelejet skal ligge. Jo mindre tallet er, jo højere ligger tonelejet.
Det tredje tal er til at vælge, hvor lang tid, lyden skal vare – målt i hundrededele af et sekund.
Det fjerde tal vælger, hvor højt tonen skal lyde. For det meste bruger man tallene fra Ø til 7, hvor 7 er det højeste. Hvis du udelader dette tal, antager maskinen, at du mener 4.

Prøv selv

Skriv **sound 1,2Ø25,3ØØ,3** og tryk på **ENTER.**

I denne "Prøv selv" øvelse spilles en dyb, lang tone med lav styrke.

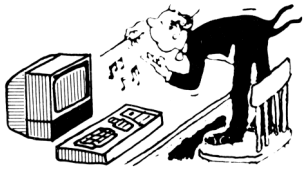
Prøv selv

Skriv **sound 1,3Ø,1Ø,7** og tryk på **ENTER.**

I denne "Prøv selv" øvelse spilles en høj, kort tone med stor styrke.

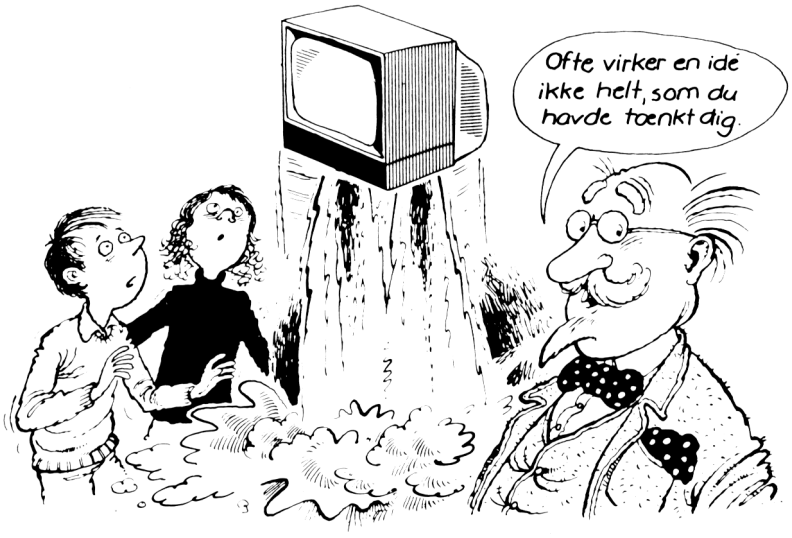
Prøv selv

Brug **sound** ordren til at spille nogle toner, du selv finder på.



Du kan få maskinen til at spille toner lige efter hinanden ved at bruge : tegnet. For eksempel:
sound 1,6Ø,1ØØ:sound 1,53,1ØØ:sound 1,47,1ØØ

PROGRAMSKRIVNING



PROF. P. NITTENGRYN

Mine råd

Det gælder om at få ideer, der virker. Du bliver sikkert nødt til at tænke en del over tingene. Hvis dine ideer ikke fungerer helt, som du havde tænkt dig, forsøg da at finde årsagerne. Det kan være, at en lille bitte ændring får det til at fungere præcist, som du havde planlagt.
Det kan nu også hænde, at din idé aldrig kommer til at virke, ligegyldigt hvor meget du ændrer den. Derfor må du være forberedt på nogle gange at skulle starte helt forfra. Det er ikke let at vælge de ideer ud, som virker, men det bliver lettere, hvis du har en plan at arbejde ud fra.
Ofte fungerer en idé ikke helt, som du havde planlagt, men kan måske bruges i en anden sammenhæng. Derfor skal du skrive den ned til senere brug. Du skal selvfølgelig især lave en omhyggelig fortegnelse over dine virkelig gode ideer.

Per Nittengryn

Datamaskinen kan gøre mange forskellige ting. Her er nogle få af dem. Prøv dem selv.

cls (efterfulgt af **ENTER**) visker skærmen ren.

print (efterfulgt af **ENTER**) giver en blank linje.

print "datamaskine" (efterfulgt af **ENTER**) udskriver **datamaskine.**

print "programmør" (efterfulgt af **ENTER**) udskriver **programmør.**

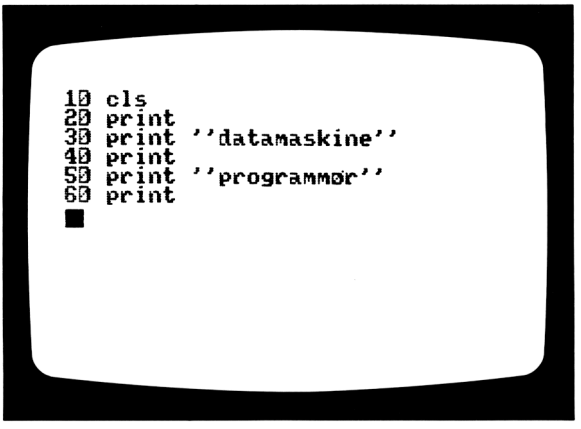
Man kan give maskinen ordrer til at udføre mange andre ting, men i stedet for at give den en ordre ad gangen, kan man afgive dem samlet. Når man giver maskinen en liste over ting, den skal udføre i en bestemt rækkefølge, kaldes det et program. Linjerne skal starte med et linjenummer for at fortælle maskinen i hvilken rækkefølge, linjerne skal udføres. Et eksempel på linjenumre er: **1Ø 2Ø 3Ø 4Ø 5Ø 6Ø** o.s.v.

Datamaskinen kigger på listen og arbejder sig gennem den fra **1Ø, 2Ø, 3Ø, 4Ø, 5Ø, 6Ø** i rækkefølge, indtil den kommer til den sidste linje. Nogle programmer består kun af nogle få linjer, andre måske af hundredevis eller endda tusindvis. Et eksempel på et program er:

1Ø cls
2Ø print
3Ø print "datamaskine"
4Ø print
5Ø print "programmør"
6Ø print

Du er begyndt at studere - kunsten at programmere.

Sådan ser det ud på skærmen:



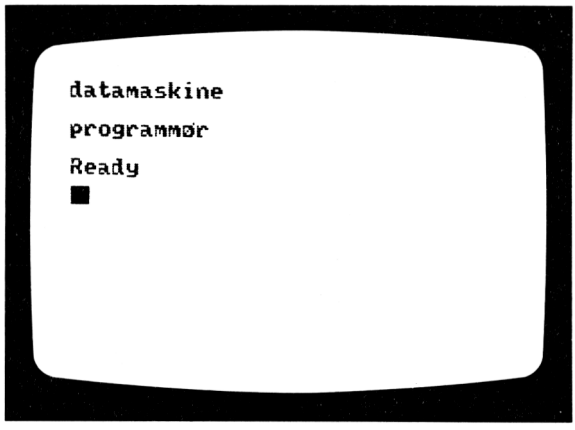
Prøv selv

Skriv programmet ovenover linje for linje. Når du har skrevet linjen færdig, tryk da på **ENTER** for at komme til begyndelsen af næste linje.

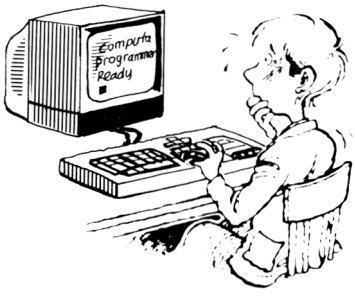
Prøv selv

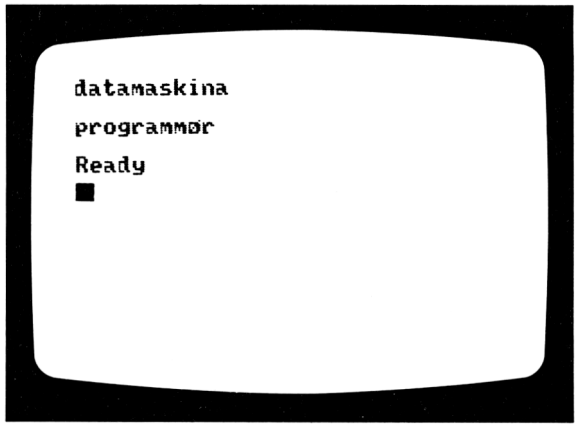
Skriv nu ordet **run** og tryk på **ENTER**.

Sådan ser det ud på skærmen:



Hvis du har lavet nogle fejl, kan du rette dem ved at skrive den linje om, som de står på. Måske viser din skærm:





Fejlen er i linje nummer **3Ø** og den kan så rettes ved at skrive linje **3Ø** om.
3Ø print "datamaskine"
(efterfulgt af **ENTER**).
Dette vil rette programlinjen og give det rigtige skærmbillede.
På samme måde kan du også lave ændringer i programlinjer.
Skriv det følgende:
3Ø print "en datamaskine"
(efterfulgt af **ENTER**).
Dette vil ændre programlinjen.
Mens du sidder og skriver et program, kan det være en hjælp at kigge tilbage i de programlinjer, du allerede har skrevet. For at gøre dette skriver man ordet **list** efterfulgt af **ENTER.**

Sådan ser det ud på skærmen:

For at finde ting, man har glemt - man bruger list, det er så nemt

```
Ready
list
10 CLS
20 PRINT
30 PRINT 'en datamaskine'
40 PRINT
50 PRINT 'programmør'
60 PRINT
Ready
```

Læg mærke til, at nogle af ordene nu står med stort. Dette viser, at datamaskinen har genkendt dem som BASIC ordrer. (Det er ligegyldigt, om du skriver dem med store eller små bogstaver).
Der er ti tal mellem hvert linjenummer for, at man skal kunne tilføje ekstra linjer. Prøv for eksempel at skrive:
15 print "jeg er" (efterfulgt af **ENTER**).
Dette indsætter linjen i programmet på følgende måde:

1Ø cls
15 print "jeg er"
2Ø print
3Ø print "en datamaskine"
4Ø print
5Ø print "programmør"
6Ø print
Du kan altid kontrollere, at linjen er rigtigt indsat ved at skrive **list** efterfulgt af **ENTER.**
Skriv **run** efterfulgt af **ENTER.**

Skriv ned

1. Giv hver linje numrene **1Ø, 2Ø, 3Ø** o.s.v.
2. Tryk på **ENTER**, når linjen er færdigskrevet.
3. Skriv **run** og tryk på **ENTER**, når hele programmet er færdigt.
4. Ret fejl ved at skrive linjen om.
5. Skriv **list** og tryk på **ENTER** for at se en udskrift af programmet.

Sådan ser det ud på skærmen:

```
Jeg er
en datamaskine
programmør
Ready
```

Dine egne programmer

For selv at kunne skrive programmer, må du finde ud af, hvad datamaskinen rent faktisk kan udføre. Det kan de næste øvelser hjælpe dig med.

Før du begynder på næste afsnit, er det nødvendigt, at du har forstået det forrige afsnit. Hvis der er noget, du er usikker på, forsøg da at gå det igennem igen.

Skriv disse programmer på samme måde som de tidligere; altså en linje ad gangen efterfulgt af **ENTER** for at komme til begyndelsen af næste linje. Når du har skrevet hele dit program, skal du kigge det efter og rette fejlene ved at skrive de linjer om, der var fejl i. Når du er sikker på, at programmet er helt færdigt og fejlfrit, skriver du **run,** trykker på **ENTER** og ser, hvad der sker.

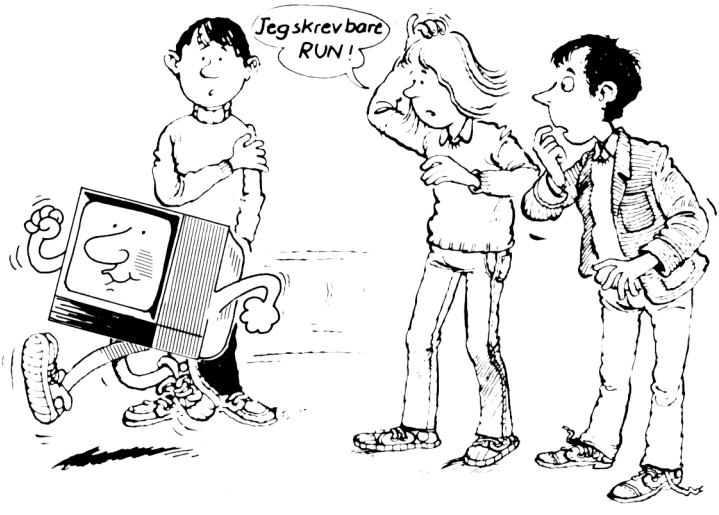
For at undgå, at maskinen sammenblander de forskellige programmer, skriver man **new** og trykker på **ENTER,** før man taster et nyt program ind.

Datamaskinen glemmer så det gamle program.

Når du har indtastet og afprøvet programmerne, prøv da at lave øvelserne. Hvis du får en eller anden idé til lignende programmer, skal du afprøve den og se, hvordan den fungerer.

I slutningen af dette kapitel er der fem opgaver, som giver nogle ideer til længere og mere interessante programmer. I de fleste tilfælde bliver du hjulpet med at skrive de grundlæggende programlinjer og bliver så bagefter forsynet med nogle ideer til forbedringer af programmet.

Skriv RUN for at se -
det virker måske.



Programmer, der bruger PRINT

Amstrad Bruger-vejledning s. G2.4.

PRINT bruges til at få udskrevet lister, information, diagrammer og ordrer.

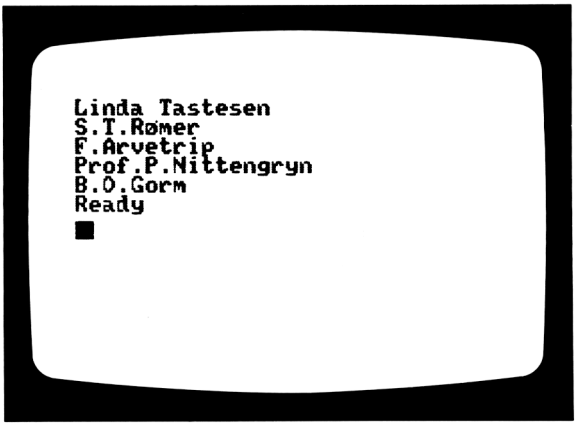
Prøv selv

Skriv **new** og tryk på **ENTER.**
Tast det følgende program ind:

```
10 cls
20 print "Linda Tastesen"
30 print "S. T. Rømer"
40 print "F. Arvetrip"
50 print "Prof. P. Nittengryn"
60 print "B. O. Gorm"
```

Skriv **run** og tryk på **ENTER.**

Sådan ser det ud på skærmen:



Når du starter på et nyt program, skal du huske først at skrive NEW.

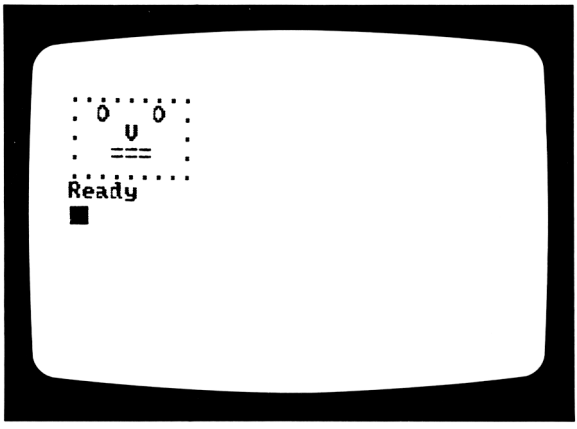
Prøv selv

Skriv **new** og tryk på **ENTER.**
Tast det følgende program ind:

```
10 cls
20 print ".........."
30 print ". O  O ."
40 print ".  V  ."
50 print ". === ."
60 print ".........."
```

Skriv **run** og tryk på **ENTER.**

Sådan ser det ud på skærmen:



Prøv selv

Skriv selv nogle programmer, hvor du for eksempel bruger **PRINT** til at lave følgende:

1. Skriv en indkøbsliste.
2. Lav et motorvejskilt.
3. Tegn et rumskib.

Programmer, der bruger TAB

Amstrad Bruger-vejledning kap.3, s.6.

TAB bruges til at opstille information eller diagrammer.

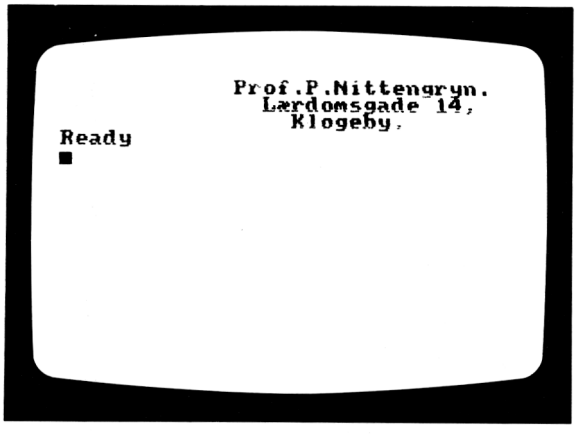
Prøv selv

Skriv **new** og tryk på **ENTER.** Tast det følgende program ind:

```
10 cls
20 print tab(20) "Prof.
P. Nittengryn,"
30 print tab(22) "Lærdomsgade 14,"
40 print tab(24) "Klogeby."
```

Skriv **run** og tryk på **ENTER**.

Sådan ser det ud på skærmen:



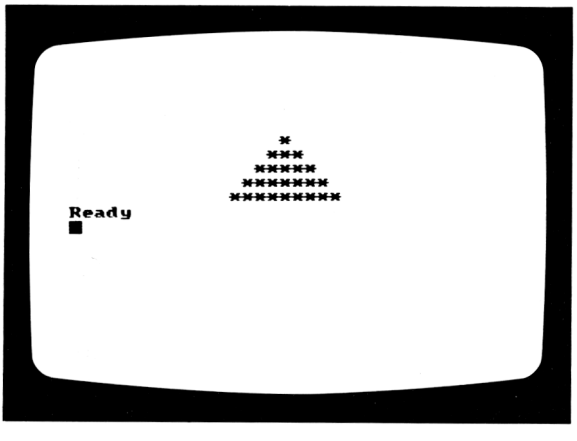
Prøv selv

Skriv **new** og tryk på **ENTER**. Tast det følgende program ind:

```
10 cls
20 print tab(19) "*"
30 print tab(18) "***"
40 print tab(17) "*****"
50 print tab(16) "*******"
60 print tab(15) "*********"
```

Skriv **run** og tryk på **ENTER**.

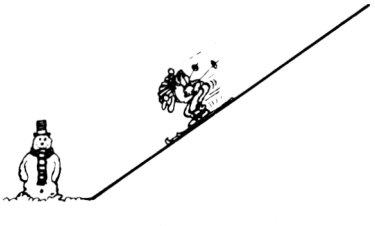
Sådan ser det ud på skærmen:



Prøv selv

Skriv dine egne programmer, hvor du bruger **TAB** til for eksempel følgende:

1. Tegn en trappe.
2. Opstil din egen adresse.
3. Tegn en skrå linje.



Programmer, der bruger LOCATE

Amstrad Bruger-vejledning s. G3.7.

LOCATE bruges til at skrive på skærmen, hvor man vil.

Prøv selv

Skriv **new** og tryk på **ENTER**. Tast det følgende program ind:

```
10 cls
20 locate 25,5
30 print "Her er jeg."
40 locate 15,23
50 print "Jeg flyttede mig."
60 locate 5,1
70 print "Halløøøøj!"
```

Skriv **run** og tryk på **ENTER**.

Sådan ser det ud på skærmen:

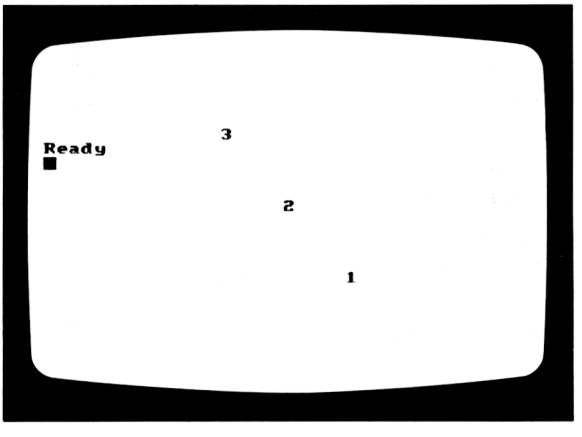
Prøv selv

Skriv **new** og tryk på **ENTER**. Indtast det følgende program:

```
10 cls
20 locate 25,17
30 print 1
40 locate 20,12
50 print 2
60 locate 15,7
70 print 3
```

Skriv **run** og tryk på **ENTER**.

Sådan ser det ud på skærmen:



Prøv selv

Lav selv nogle programmer, hvor du benytter **LOCATE** til for eksempel følgende:

1. Udskriv beskeder på forskellige dele af skærmbilledet.
2. Tegn en takket linje.

Programmer, der bruger =

= tegnet benyttes til at lægge et tal ind på en adresse.

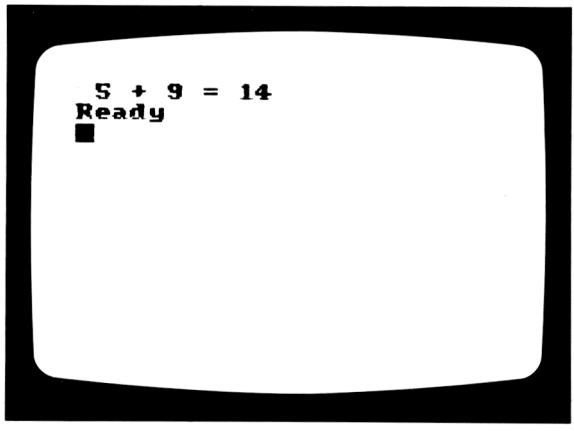
Prøv selv

Skriv **new** og tryk på **ENTER**. Indtast det følgende program:

```
10 cls
20 a=5
30 b=9
40 sum=a+b
50 print a;"+";b;"=";sum
```

Skriv **run** og tryk på **ENTER**.

Sådan ser det ud på skærmen:



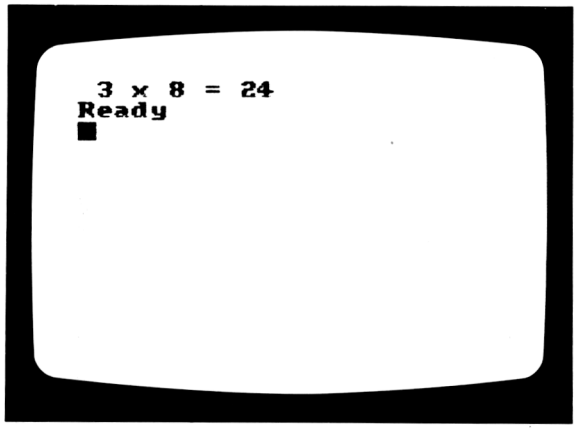
Prøv selv

Skriv **new** og tryk på **ENTER**. Indtast det følgende program:

```
10 cls
20 e=3
30 f=8
40 resultat=e*f
50 print e;"x";f;"=";resultat
```

Skriv **run** og tryk på **ENTER**.

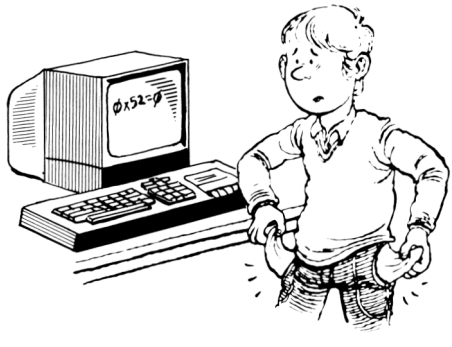
Sådan ser det ud på skærmen:



Prøv selv

Lav selv nogle programmer, der bruger = til at kunne:

1. Lægge to tal sammen.
2. Gange to andre tal.
3. Trække fra eller dividere nogle nye tal.



= tegnet bruges også til at lægge ord ind på adresser.

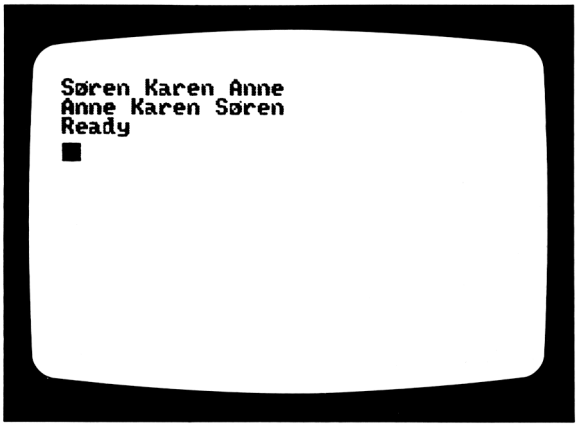
Prøv selv

Skriv **new** og tryk på **ENTER**. Indtast det følgende program:

```
10 cls
20 a$="Karen"
30 b$="Søren"
40 c$="Anne"
50 print a$" "b$" "c$
60 print c$" "b$" "a$
```

Skriv **run** og tryk på **ENTER**.

Sådan ser det ud på skærmen:



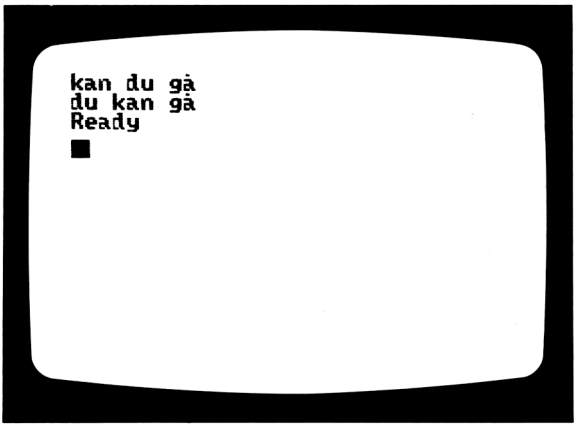
Prøv selv

Skriv **new** og tryk på **ENTER**. Tast det følgende program ind:

```
10 cls
20 a$="kan"
30 b$="du"
40 c$="gå"
50 print a$" "b$" "c$
60 print b$" "a$" "c$
```

Skriv **run** og tryk på **ENTER**.

Sådan ser det ud på skærmen:



Prøv selv

Forsøg at lave nogle programmer, hvor du bruger = tegnet. For eksempel til dette:

1. Sæt **a$="frem", b$="lige", c$="tid", d$="mål", e$="rum", f$="for", g$="mand".**

Prøv ved at sætte disse ord sammen at se, hvor mange længere ord, du kan skrive.

2. Lav sætninger af ord.
3. Vælg nogle punkter fra en oversigt.

Programmer, der bruger INPUT

Amstrad Bruger-vejledning s. G2.6.

INPUT bruges til at sætte tal ind i programmer.

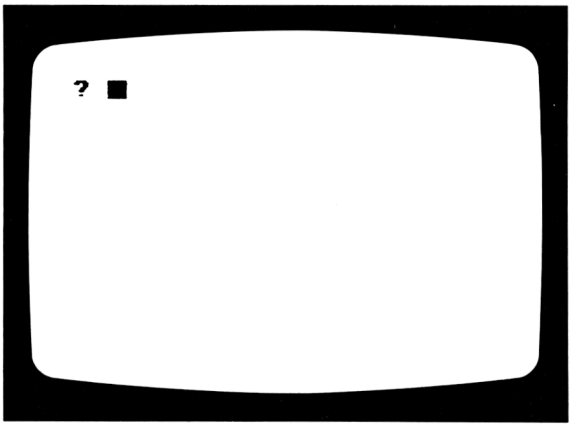
Prøv selv

Skriv **new** og tryk på **ENTER**. Indtast det følgende program:

```
10 cls
20 input a
30 input b
40 forskel=a-b
50 print a;"-";b;"=";forskel
```

Skriv **run** og tryk på **ENTER**.

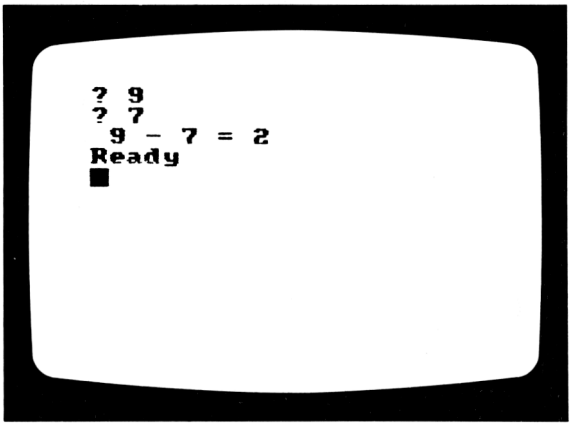
Sådan ser det ud på skærmen:



Prøv selv

Tast et tal ind, for eksempel **9**, og tryk på **ENTER**. Tast herefter et andet tal ind, for eksempel **7**, og tryk på **ENTER**.

Sådan ser det ud på skærmen:



Prøv selv

Prøv nu programmet med dine egne tal.

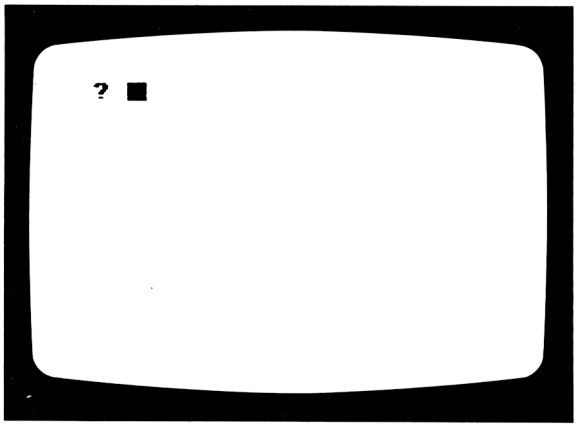
Prøv selv

Skriv **new** og tryk på **ENTER**. Indtast det følgende program:

```
10 cls
20 input t
30 input n
40 deling=t/n
50 print t;"/";n;"=";deling
```

Skriv **run** og tryk på **ENTER**.

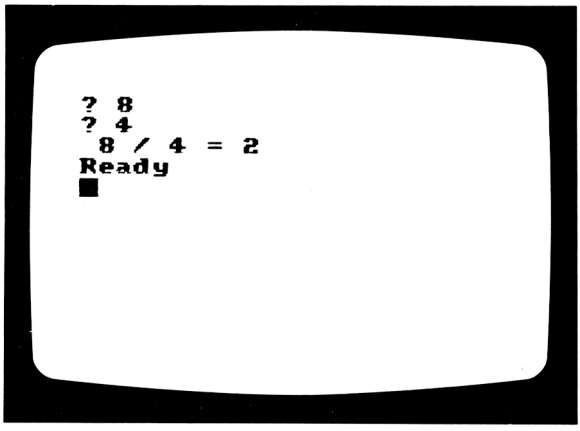
Sådan ser det ud på skærmen:



Prøv selv

Tast et tal ind, for eksempel **8**, og tryk på **ENTER**. Tast et andet tal ind, for eksempel **4** og tryk på **ENTER**.

Sådan ser det ud på skærmen:



Prøv selv

Skriv **run** og tryk på **ENTER**.
Sæt nu nogle tal, du selv finder på, ind i programmet.

Prøv selv

Lav selv nogle programmer, der bruger **INPUT**, for eksempel til følgende:

1. Læg to tal sammen.
2. Gang to tal med hinanden.
3. Lad programmet spørge om beløbet af dine ugepenge og gang dette med 52 for at komme frem til det årlige beløb.

INPUT bruges også til at sætte ord ind i programmer.

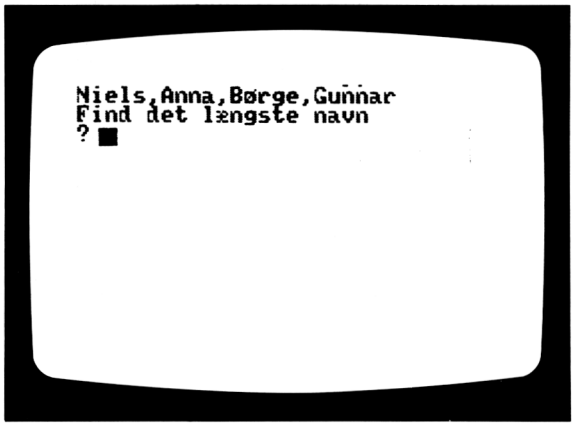
Prøv selv

Skriv **new** og tryk på **ENTER**. Tast det følgende program ind:

```
10 cls
20 print "Niels,Anna,Børge,Gunnar"
30 print "Find det længste navn."
40 input svar$
50 print "Det længste navn er Gunnar"
```

Skriv **run** og tryk på **ENTER**.

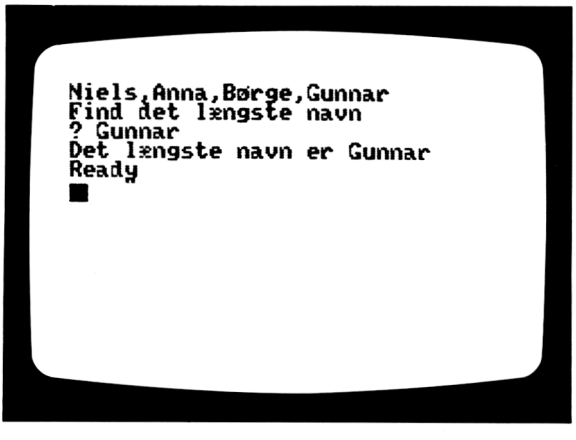
Sådan ser det ud på skærmen:



Prøv selv

Indtast det længste navn **Gunnar** og tryk på **ENTER**.

Sådan ser det ud på skærmen:



Prøv selv

Skriv **new** og tryk på **ENTER**. Indtast dette program:

```
10 cls
20 print "Giv mig et ord, der passer
til denne sætning"
30 print "I dag er vejret"
40 input svar$
50 print "Vejret er";svar$
```

Sådan ser det ud på skærmen:

```
Giv mig et ord,der passer
til denne sætning

I dag er vejret
?
```

Når en linje er for lang til at stå på skærmen, skal du bare blive ved med at skrive videre.
Maskinen forstår dig nemlig godt, selv om et ord bliver delt på to linjer.
Du skal ikke trykke på **ENTER**, før du har skrevet hele sætningen færdig.
Skriv **run** og tryk på **ENTER**.

Prøv selv

Indtast **godt** (eller et andet ord, der beskriver vejret) og tryk på **ENTER**.

Sådan ser det ud på skærmen:

```
Giv mig et ord,der passer
til denne sætning

I dag er vejret
? godt
I dag er vejret godt
Ready
```

Prøv selv

Find selv på nogle programmer, hvor du bruger **INPUT**, for eksempel til dette:

1. Finde et ord, der mangler.
2. Svare på et let spørgsmål.
3. Få datamaskinen til at føre en samtale med dig.

Programmer, der bruger FOR/TO/STEP/NEXT

Amstrad Bruger-vejledning kap. 8, s. 18.

FOR/TO/STEP/NEXT bruges til at gentage de samme linjer for hvert tal i en talrække.

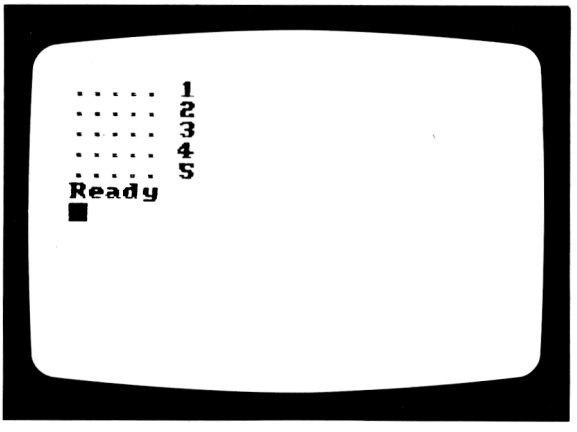
Prøv selv

Skriv **new** og tryk på **ENTER**. Indtast dette program:

```
10 cls
20 for tæl=1 to 5 step 1
30 print ".....";tæl
40 next tæl
```

Skriv **run** og tryk på **ENTER**.

Sådan ser det ud på skærmen:



For hver gang **tæl** bliver tildelt værdi mellem 1 og 5, bliver rækken af prikker og værdien af **tæl** udskrevet.

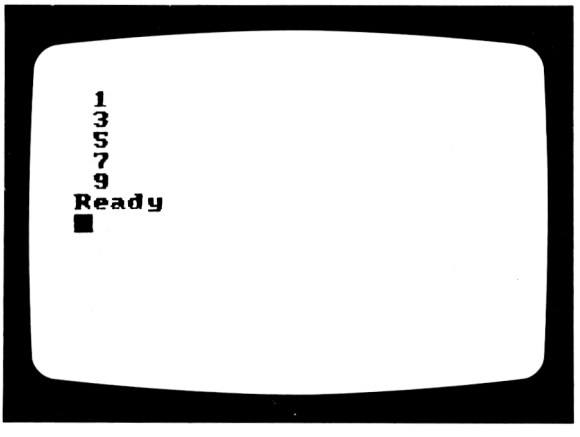
Prøv selv

Skriv **new** og tryk på **ENTER**. Indtast dette program:

```
10 cls
20 for t=1 to 9 step 2
30 print t
40 next t
```

Skriv **run** og tryk på **ENTER**.

Sådan ser det ud på skærmen:



Prøv selv

Forsøg selv at lave nogle programmer, hvor du benytter **FOR/TO/STEP/NEXT** til for eksempel:

1. Udskriv tallene fra 1 til 20.
2. Udskriv dit navn og din adresse tre gange.
3. Udskriv tretabellen.

Hvis du ikke fortæller maskinen, hvor meget den skal tælle op ad gangen, tæller den 1 op hver gang. Derfor kunne vi godt have udeladt **step 1** i det første eksempel.

Programmer, der bruger DATA/READ

Amstrad Bruger-vejledning kap. 8, s. 9 og s. 37.

DATA/READ bruges til at lægge data (information) ind i datamaskinen og indlæse dem (kalde dem frem), når man har brug for dem.

Prøv selv

Skriv **new** og tryk på **ENTER**. Indtast dette program:

```
10 cls
20 data er,spor,et,her
30 read w$,x$,y$,z$
40 print z$;" ";w$;" ";y$;" ";x$
50 print y$;" ";x$;" ";w$;" ";z$
```

Skriv **run** og tryk på **ENTER**.

Sådan ser det ud på skærmen:

```
her er et spor
et spor er her
Ready
```

Prøv selv

Skriv **new** og tryk på **ENTER**. Indtast det følgende program, hvor fem elevers matematik og engelsk karakterer indlæses og deres gennemsnit beregnes.

```
10 cls
20 data 7,11,8,5,3,13,
9,7,9,6
30 for elev=1 to 5
40 read matematik, engelsk
50 gennemsnit=(matematik+
engelsk)/2
60 print "elev";elev;"har et
gennemsnit på"; gennemsnit
70 next elev
```

Skriv **run** og tryk på **ENTER**.

Sådan ser det ud på skærmen:

```
elev 1 har et gennemsnit på 9
elev 2 har et gennemsnit på 6.5
elev 3 har et gennemsnit på 8
elev 4 har et gennemsnit på 8
elev 5 har et gennemsnit på 7.5
Ready
```

Prøv selv

Lav dine egne programmer, hvor du benytter **DATA/READ** til for eksempel dette:

1. Lav om på rækkefølgen af nogle punkter på en liste.
2. Lav en liste over månederne, der viser antallet af dage i hver måned.

Programmer, der bruger LEFT$, MID$ og RIGHT$

Amstrad Bruger-vejledning kap. 8, s. 23,27 og 40.

Ordfunktionerne **LEFT$, MID$** og **RIGHT$** kan benyttes til at udskille dele af ord.

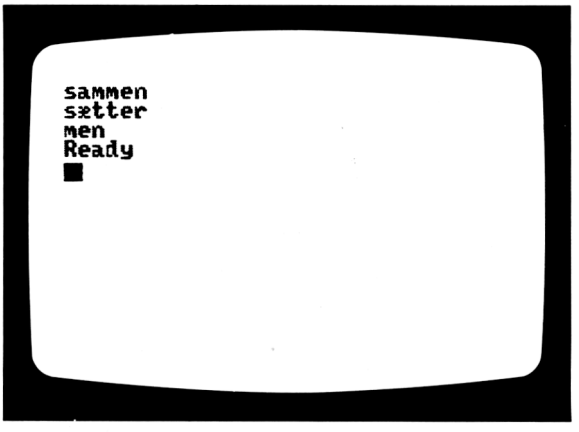
Prøv selv

Skriv **new** og tryk på **ENTER**. Indtast følgende program:

```
10 cls
20 ord$="sammensætter"
30 print left$(ord$,6)
40 print right$(ord$,6)
50 print mid$(ord$,4,3)
```

Skriv **run** og tryk på **ENTER**. Brug **LEFT$, MID$** og **RIGHT$** til at udskrive så mange ord, du kan finde på, hvis **ord$="båndoptager"**.

Sådan ser det ud på skærmen:



Programmer, der bruger INT og RND

Amstrad Bruger-vejledning kap. 8, s. 22 og 40.

INT giver heltalsdelen af et tal. For eksempel er **INT(2.4)=2. RND** vælger et tilfældigt tal mellem Ø og 1.

Prøv selv

Skriv **new** og tryk på **ENTER**. Indtast følgende program:

```
10 cls
20 for t=1 to 20
30 print t/4,int(t/4)
40 next t
```

Skriv **run** og tryk på **ENTER**.

Dette program viser, hvordan **INT** virker. Man er nødt til at anvende **INT** sammen med **RND,** hvis man vil have hele tal som svar. **RND** vælger et tal mellem Ø og 1 (men ikke tallet 1). **RND*4** vælger således et tal mellem Ø og 4 (men ikke tallet 4). **INT(RND*4)** vælger et af tallene Ø, 1, 2 eller 3.

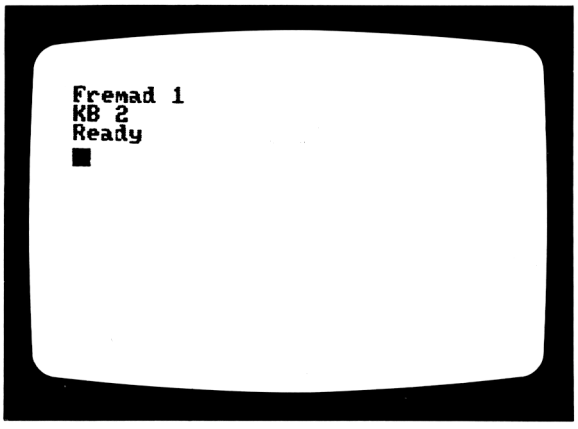
Prøv selv

Skriv **new** og tryk på **ENTER**. Tast dette program ind:

```
10 cls
20 Fremad=int(rnd*4)
30 KB=int(rnd*4)
40 print "Fremad";Fremad
50 print "KB";KB
```

Skriv **run** og tryk på **ENTER**.

Sådan kan det se ud på skærmen:



Da datamaskinen vælger tilfældige tal, får du sandsynligvis ikke de samme tal på din skærm, som dem der vises i denne og den næste "Prøv selv".

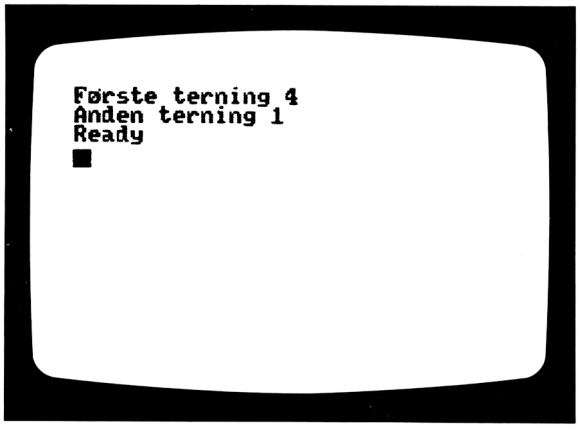
Prøv selv

Skriv **new** og tryk på **ENTER**. Indtast følgende:

```
10 cls
20 første=1+int(rnd*6)
30 anden=1+int(rnd*6)
40 print "Første terning";første
50 print "Anden terning";anden
```

Skriv **run** og tryk på **ENTER**.

Sådan kan det se ud på skærmen:



Ordren **første=1+int(rnd*6)** vælger tilfældigt et af tallene 1, 2, 3, 4, 5 eller 6, og lægger det valgte tal ind på adressen **første.** Prøv at udføre programmet flere gange, hvor forskellige tal sikkert bliver valgt for hver gang.

Prøv selv

Forsøg selv at lave nogle programmer med **INT** og **RND** til for eksempel følgende:

1. Udtræk en vinderlodseddel ud af 1000 sedler.
2. Udvælg nogle tal til et spil bingo (tal fra 1 til 90).
3. Læg nogle tal sammen og lad det rigtige resultat udskrive, når du har gjort et forsøg på at svare.

Programmer, der bruger WHILE/WEND

Amstrad Bruger-vejledning kap. 8, s. 51.

WHILE/WEND benyttes til at gentage programlinjer, når en særlig betingelse opfyldes.

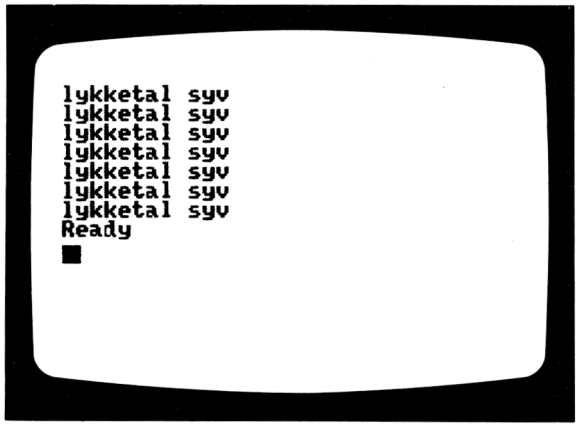
Prøv selv

Skriv **new** og tryk på **ENTER**. Indtast dette:

```
10 cls
20 tæl=0
30 while tæl < 7
40 print "lykketal syv"
50 tæl=tæl+1
60 wend
```

Skriv **run** og tryk på **ENTER**.

Sådan ser det ud på skærmen:



Den særlige betingelse, der skal være opfyldt i denne "Prøv selv" er, at **tæl** < 7 (< betyder **mindre end**). Programmet gentager linjerne 40 og 50, så længe **tæl** er mindre end 7.

Den næste "Prøv selv" bruger to nye ideer. Den første er en metode til at sætte to ord sammen, hvilket man gør ved at skrive + mellem dem. Den anden nye idé er en metode til at finde antallet af tegn i et ord, for eksempel **x$**, ved at skrive **LEN(x$)**. Hvis **x$="Prof."**, er **LEN(x$)=5**. **LEN** fortæller, hvor mange pladser, et ord optager på skærmen.

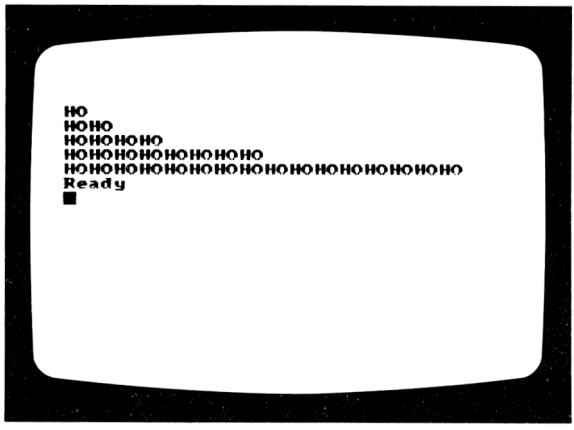
Prøv selv

Skriv **new** og tryk på **ENTER**. Tast dette ind:

```
10 cls
20 b$="HO"
30 while len(b$)<33
40 print b$
50 b$=b$+b$
60 wend
```

Skriv **run** og tryk på **ENTER**.

Sådan ser det ud på skærmen:



Lav dine egne programmer, der bruger **WHILE/WEND**, for eksempel til følgende:

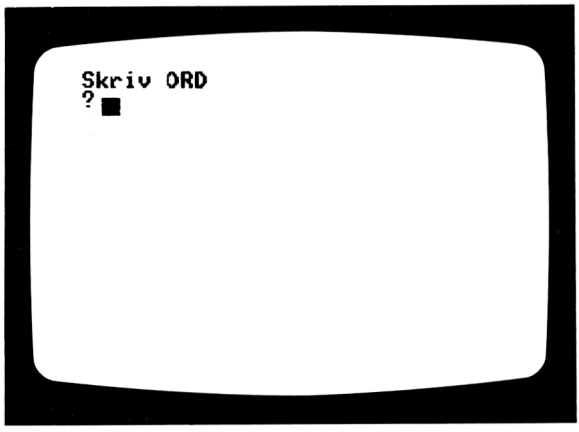
1. Udskriv tallene fra 1 til 20.
2. Udskriv spørgsmålet "Hvad er 10x44" og indlæs svarene, indtil det rigtige kommer.

Programmer, der bruger IF/THEN/ELSE

Amstrad Bruger-vejledning kap. 8, s. 19.

IF/THEN/ELSE bruges til at vælge mellem flere forskellige muligheder.

Sådan ser det ud på skærmen:



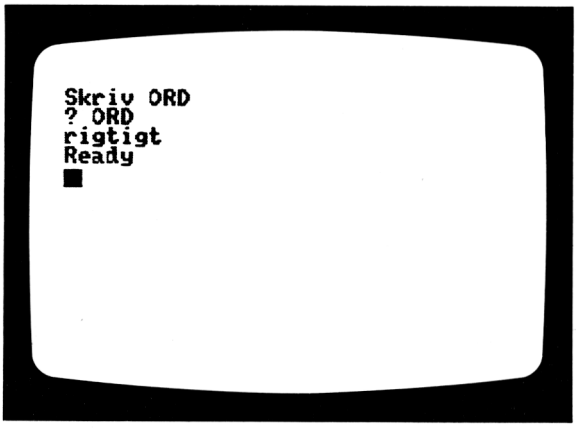
Prøv selv

Skriv **new** og tryk på **ENTER**. Indtast dette program:

```
10 cls
20 print "Skriv ORD"
30 input ord$
40 if ord$="ORD" then print
"rigtigt" else print "forkert"
```

Skriv **run** og tryk på **ENTER**.

Sådan ser det ud på skærmen:

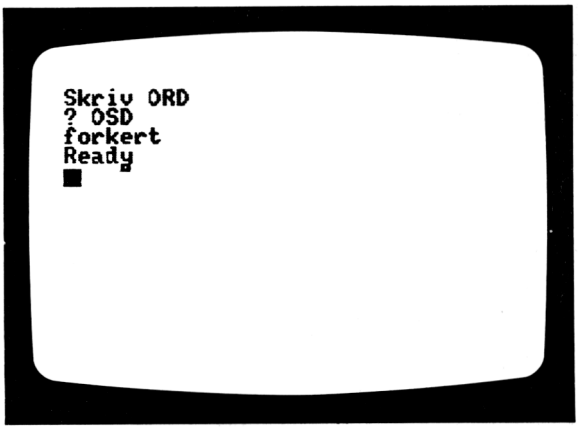


Prøv selv

Skriv **ORD** og tryk på **ENTER**.

Læg mærke til, at du bliver nødt til at skrive **ORD** med store bogstaver for at få det til at fungere. Lad maskinen udføre programmet igen.

Sådan ser det ud på skærmen:



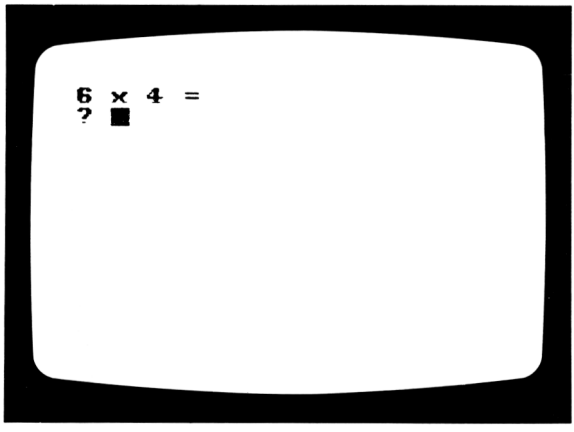
Prøv selv

Skriv **OSD** og tryk på **ENTER**.

Prøv selv

Skriv **new** og tryk på **ENTER**. Indtast dette:
1Ø cls
2Ø print "6×4="
3Ø input svar
4Ø if svar=24 then print "rigtigt" else print "forkert"
Skriv **run** og tryk på **ENTER**.

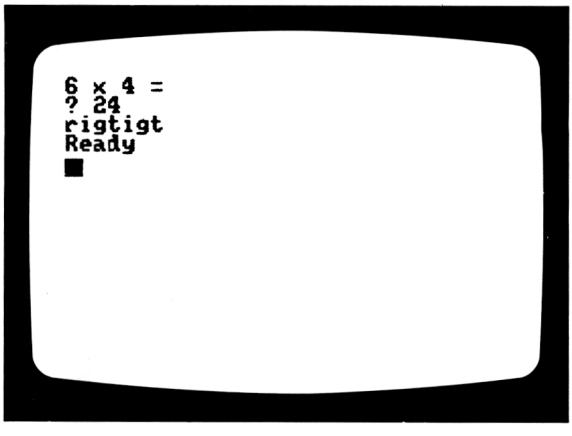
Sådan ser det ud på skærmen:



Prøv selv

Skriv **24** og tryk på **ENTER**.

Sådan ser det ud på skærmen:

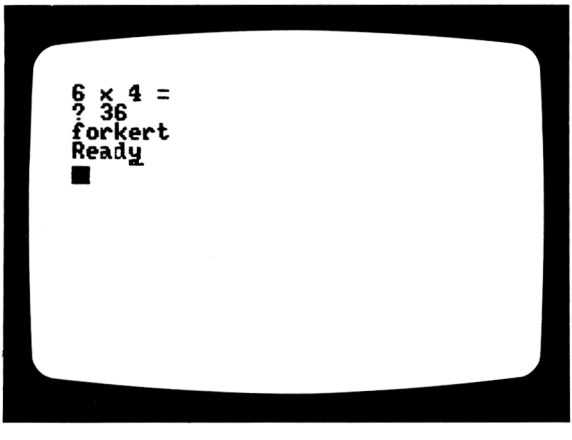


Lad maskinen udføre programmet igen.

Prøv selv

Skriv **36** og tryk på **ENTER**.

Sådan ser det ud på skærmen:



Prøv selv

Forsøg at lave nogle programmer, hvor du bruger **IF/THEN/ELSE** til at løse følgende opgaver:
1. Kontrollér svaret på et let spørgsmål.
2. Få maskinen til at spørge, om du har det godt og derpå give en passende bemærkning til dit svar.

Programmer, der bruger GOTO

Amstrad Bruger-vejledning s. G2.5.

GOTO bruges til at hoppe til en anden linje i programmet.

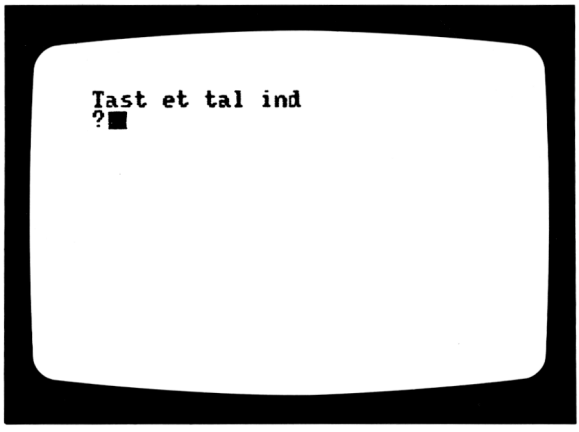
Prøv selv

Skriv **new** og tryk på **ENTER**. Tast dette program ind:

1Ø cls
2Ø print "Tast et tal ind"
3Ø input tal
4Ø if tal < 1ØØ then goto 7Ø
5Ø print tal "er større end 99"
6Ø end
7Ø print tal "er mindre end 1ØØ"

Skriv **run** og tryk på **ENTER**.

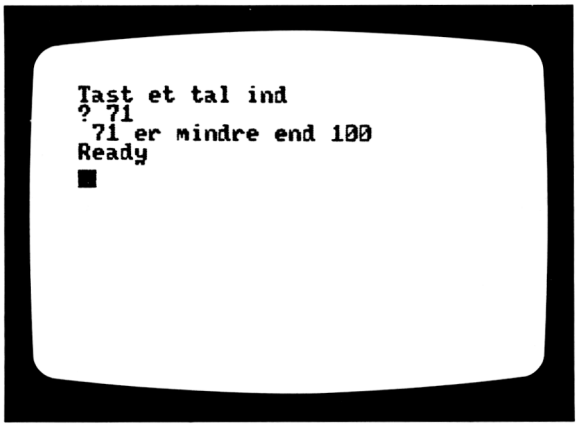
Sådan ser det ud på skærmen:



Prøv selv

Tast nu et tal ind, for eksempel **71**, og tryk på **ENTER**.

Sådan ser det ud på skærmen:



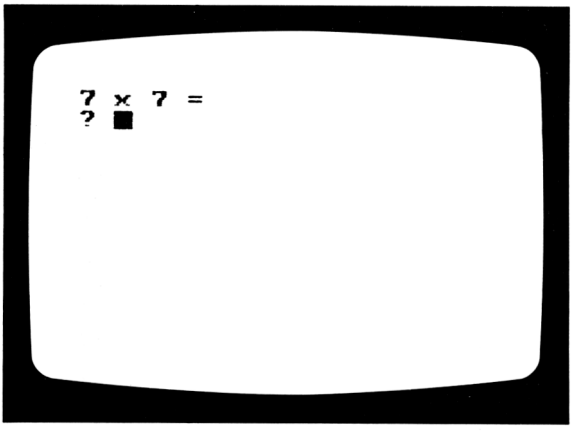
Prøv selv

Skriv **new** og tryk på **ENTER**. Tast dette program ind:

```
10 cls
20 print "7×7="
30 input svar
40 if svar=49 then goto 70
50 print "forkert"
60 goto 20
70 print "rigtigt"
```

Skriv **run** og tryk på **ENTER**.

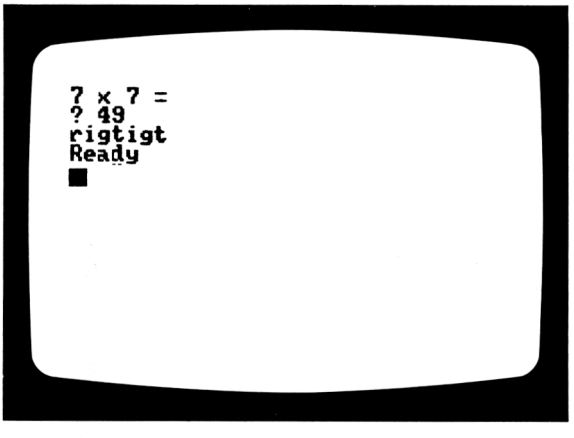
Sådan ser det ud på skærmen:



Prøv selv

Indtast dit svar og tryk på **ENTER**.

Sådan ser det ud på skærmen:



Prøv selv

Find selv på nogle programmer, hvor du bruger **GOTO**, for eksempel med disse emner:

1. Udskriv **ja** eller **nej** som svar på et spørgsmål.
2. Udskriv et ord med små eller store bogstaver.
3. Lad priserne på fem indkøbsvarer være input i et program og udskriv det samlede beløb. Du må sørge for, at maskinen hele tiden holder styr på, hvor mange varer, du har læst ind.

Programmer, der bruger AND

AND benyttes til at kontrollere to betingelser.

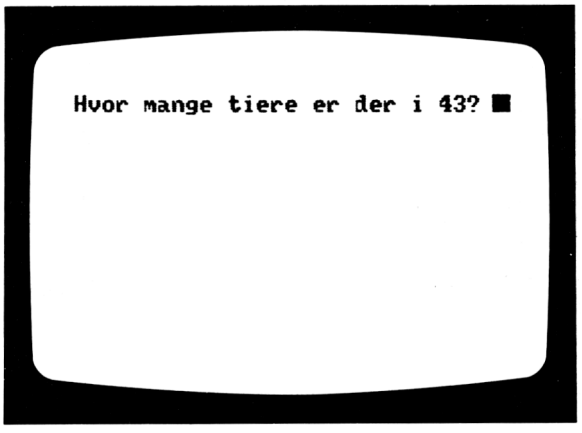
Prøv selv

Skriv **new** og tryk på **ENTER**. Indtast dette:

```
10 cls
20 print "Hvor mange tiere er der i 43";
30 input tiere
40 print "Hvor mange enere er der i 43";
50 input enere
60 if tiere=4 and enere=3 then print "Begge svar rigtige" else goto 20
```

Skriv **run** og tryk på **ENTER**.

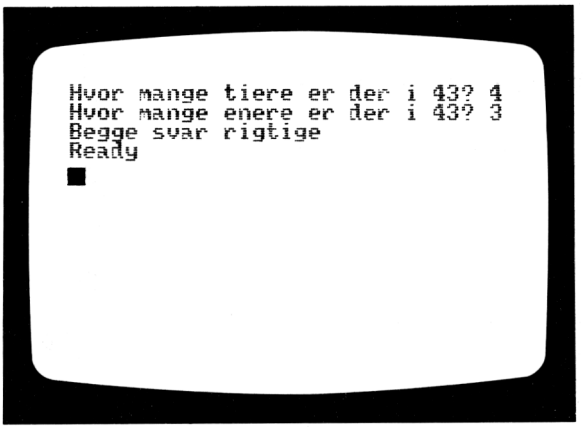
Sådan ser det ud på skærmen:



Prøv selv

Tast **4** ind og tryk på **ENTER**.
Tast **3** ind og tryk på **ENTER**.

Sådan ser det ud på skærmen:



Prøv selv

Lad programmet udføres igen.
Skriv et eller to forkerte svar og se, hvad der sker.

Prøv selv

Skriv **new** og tryk på **ENTER**. Tast dette program ind:

1Ø cls
2Ø print "Hvad er det første bogstav i alfabetet"
3Ø input a$
4Ø print "Hvad er det sidste bogstav i alfabetet"
5Ø input å$
6Ø if a$="a" and å$="å" then print "Begge svar rigtige" else goto 2Ø

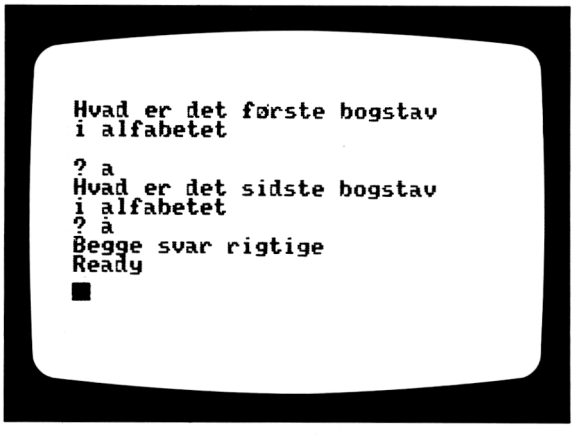
Skriv **run** og tryk på **ENTER**.

Sådan ser det ud på skærmen:

Prøv selv

Tast **a** ind og tryk på **ENTER**.
Tast **å** ind og tryk på **ENTER**.

Sådan ser det ud på skærmen:



Prøv selv

Lad maskinen udføre programmet igen.
Tast et eller to forkerte bogstaver ind og se, hvad der sker.

Prøv selv

Find på dine egne programmer, der bruger **AND** til at udføre for eksempel følgende:

1. Find på et spørgsmål med to mulige svar og kontrollér disse.
2. Vælg to ord ud fra en liste over nogle ord.
3. Vælg to tal ud fra en liste over nogle tal.

Programmer, der bruger OR

OR bruges til at kontrollere den ene betingelse ud af to mulige.

Prøv selv

Skriv **new** og tryk på **ENTER**. Tast følgende program ind:

```
10 cls
20 print "Hvilket bogstav mangler i"
30 print "dyret H.NE"
40 input bogstav$
50 if bogstav$="A" or
bogstav$="Ø"
then print "rigtigt" else goto 20
```

Skriv **run** og tryk på **ENTER**.

Sådan ser det ud på skærmen:

Prøv selv

Skriv **A** og tryk på **ENTER**.

Sådan ser det ud på skærmen:

Lad maskinen udføre programmet igen.

Sådan ser det ud på skærmen:

Prøv selv

Skriv **Ø** og tryk på **ENTER**.

Prøv selv

Skriv nogle forkerte svar og se, hvad der sker.

Prøv selv

Skriv **new** og tryk på **ENTER**. Indtast dette program:

```
1Ø cls
2Ø print "Indtast det mærkelige bogstav"
3Ø print "A E H I O U Æ Ø Å"
4Ø input bogstav$
5Ø if bogstav$="H" or bogstav$="h"
then print "rigtigt" else goto 2Ø
```

Skriv **run** og tryk på **ENTER**.

Sådan ser det ud på skærmen:

Prøv selv

Skriv **H** og tryk på **ENTER**.

Sådan ser det ud på skærmen:

Prøv selv

Lad programmet blive udført igen og prøv at skrive **h**. Prøv nogle flere gange, hvor du taster andre bogstaver ind.

Lav selv nogle programmer, hvor du for eksempel bruger **OR** til dette:

1. Find på et spørgsmål med to mulige svar og kontrollér det ene af dem.
2. Lad for eksempel spørgsmålet dreje sig om tal, hvor to forskellige tal begge kan være det rigtige svar.

Opgave 1 – Tænk på et tal

I denne opgave skal du lave et program, hvor datamaskinen tænker på et tal mellem 1 og 1ØØ, som du så skal gætte. Når du gætter rigtigt, skal spillet stoppe. Hvis du gætter forkert, skal maskinen fortælle dig, om dine gæt var for høje eller for lave. Her er hovedtrækkene i programmet:

1. Ved at bruge **RND** kan du få maskinen til at finde på et tilfældigt tal.
   (a) **Tal=1+int(rnd*1ØØ)**. Det tal, maskinen vælger, ligger nu på adressen **tal**.

2. Brug en **PRINT** sætning til at udskrive, at maskinen nu tænker på et tal mellem 1 og 1ØØ.
   (a) Brug en **INPUT** sætning til at indlæse dit gæt.

3. Maskinen skal nu oplyse dig om størrelsen af dit gæt.
   (a) Hvis **gæt<tal**, så skriv: Du gættede for lavt.
   (b) **Hvis gæt>tal**, så skriv: Du gættede for højt.
   (c) **Hvis gæt=tal**, så brug **GOTO** punkt 5.

4. Sæt en grænse for, hvor mange gæt, du må få.
   (a) Sæt **antallet af gæt=maxgæt**.
   (b) Brug **while/wend** sammen med en **tæller** således, at du kun får lov til at gætte, sålænge **tæller<maxgæt**: **while tæller<maxgæt**
   (c) Punkterne 2 og 3 skal sættes ind i while-løkken inden **wend**.

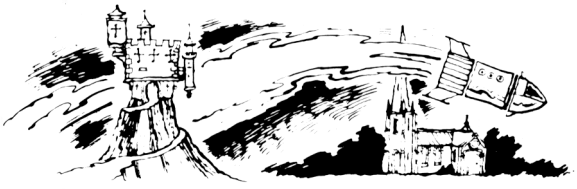
5. Brug en **PRINT** sætning til at lykønske med det rigtige gæt.

6. Hvis du vil spille en gang til med et nyt tal, må du sidst i programmet lave en **GOTO** sætning, hvor man hopper tilbage til 1(a).

Forbedringer og variationer

A. Tilføj nogle linjer i programmet således, at datamaskinen spørger, om du vil prøve igen. Hvis svaret er ja, skal den starte forfra i programmet.

B. Lav det omvendte program, hvor du tænker på et tal, som maskinen så skal gætte.



Opgave 2 – Tegning

Du kan lave en masse spændende ting ved at benytte **DRAW** og **MOVE**. Her er nogle ideer:

A. Prøv at afbilde en rumraket, en kirke, et fjernsyn, et slot, et skib eller en telefonboks. Det bliver lettere, hvis du først tegner din tegning på en kopi af planen på side 41.

B. Tegn en lille stjerne (brug planen på side 41 til hjælp). Prøv at lægge det samme tal til alle de vandrette afstande (det første tal efter **DRAW** og **MOVE**). Se, hvor stjernen nu bliver tegnet. Forsøg så at tegne stjernen længere oppe på skærmbilledet.
Prøv at bruge en **FOR...NEXT** løkke, **INT** og **RND** til at afbilde flere stjerner på tilfældige steder (således, at det bliver et nyt sted for hver gang, programmet udføres). Hvis du har en farveskærm eller et farvetv kan du forsøge at give stjernerne forskellige farver (for at gøre dette, skal du bruge **INK** ordren).

C. Forsøg at bruge:
FOR række=1 TO 39Ø STEP 3Ø: MOVE Ø, række:DRAW 639,række: NEXT række

D. Du kan tegne et kvadrat ved at bruge:
FOR række=15Ø TO 25Ø STEP 2: MOVE 15Ø,række:DRAW 539,række:NEXT række

Faktisk er det en smule højere, end det er bredt. Prøv at få det rigtigt. Forsøg at lave et rektangel, som er længere i bredden end i højden.

Opgave 3 – NIM spillet

I denne opgave skal du programmere maskinen til at spille pindespillet NIM med dig. Du begynder spillet med 2Ø pinde. Maskinen og du skiftes så til at fjerne enten 1, 2, 3 eller 4 pinde. Den, som fjerner den sidste eller de sidste pinde, vinder. Programmet kan skrives således:

1. Få maskinen til at forklare spillereglerne ved at bruge **PRINT** sætninger.

2. Start med at sætte antallet af pinde til 2Ø.

3. Når det er din tur, skal der ske følgende:
   (a) Få maskinen til at spørge om, hvor mange pinde du ønsker at fjerne og lav svaret som en **INPUT** sætning.
   (b) Træk det antal pinde fra, som du gav som svar og lad maskinen udskrive, hvor mange pinde der nu er tilbage (i en **PRINT** sætning).
   (c) Kontrollér om antallet af pinde er lig nul. I så fald skal maskinen fortælle, at du har vundet og herefter afslutte spillet.

4. Når det bliver maskinens tur, skal der ske dette:
   (a) Hvis antallet af pinde er mindre end fem (<5), kan maskinen vælge at fjerne alle pindene. Ellers skal den vælge at fjerne et tilfældigt antal mellem 1 og 4 (brug **INT(RND*4)+1**). (Brug **IF/THEN/ELSE** til denne del af programmet).
   (b) Udregn det nye antal pinde og udskriv resultatet med en **PRINT** sætning.
   (c) Kontrollér om antallet af pinde er lig nul. I så fald skal maskinen fortælle, at den er vinderen og afslutte spillet, ellers skal den gå tilbage til din tur (punkt 3). Vær sikker på, at det navn, du sætter for antallet af pinde, er det samme som i begyndelsen af punkt 3.

Forbedringer og variationer

A. Maskinen kunne spørge, om du vil begynde spillet og hvis ikke, selv begynde det.

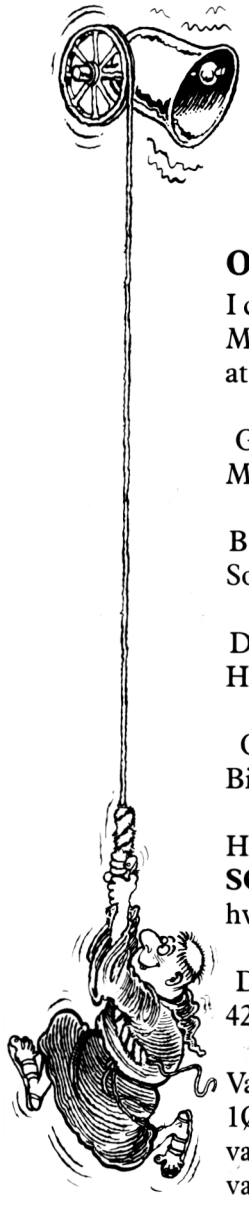
B. I stedet for, at maskinen skriver antallet af pinde, kunne den tegne dem (ved at bruge **PRINT** og bogstavet **I** for en pind).

C. Du kunne viske skærmen ren efter hver tur ved at bruge **CLS** og **PRINT** uden noget efter for at få blanke linjer og ekstra mellemrum.

D. Det ville være en god idé at lade maskinen kontrollere, at du ikke snyder ved at fjerne for mange pinde.

E. Du kunne ændre det antal pinde, man starter med og hvor mange, man må fjerne hver gang.

F. Når du har spillet et par gange og har fundet ud af, hvordan du vinder over maskinen, kan du programmere den til at spille bedre.



Opgave 4 – Musik

I dette program drejer det sig om at finde ud af at spille melodien Mester Jakob. Teksten og noderne står nedenunder (C' o.s.v. betyder, at noden ligger en oktav højere end nøgle c'et).

G A B G G A B G
Mester Jakob, Mester Jakob

B C' D' B C' D'
Sover du, sover du

D' E' D' C' B G D' E' D' C' B G
Hører du ej klokken, hører du ej klokken

G D G G D G
Bim, bam, bum. Bim, bam, bum.

Hver enkelt tone kan programmeres som en ordre, der består af **SOUND, 1, tone, varighed. Tone** står for tonelejet og **varighed** for, hvor længe tonen skal spilles. Tallene for de forskellige noder er:

D E F G A B C' D' E'
426 379 358 319 284 253 239 213 19Ø

Varigheden for de korte toner skal være 25, for de middellange 5Ø og 1ØØ for de lange toner. Hvis du synes, det er svært at finde den rette varighed, kan du først sætte alle tonerne til 5Ø og senere ændre varigheden, når du har lyttet til melodien et par gange.

Her er nogle forslag til, hvordan man kan skrive programmet.

1. Sæt **v=25** i starten af programmet (**v** står for varighed). Herefter kan du sætte varigheden af tonerne til **v**, **v*2** eller **v*4** alt efter, hvor lang tid du ønsker dem spillet. På den måde kan du let ændre på, hvor hurtigt eller langsomt, melodien spilles, blot ved at ændre værdien af **v**.

2. Du kan også prøve at bruge **DATA/READ** sætninger. Det er en bedre måde at programmere på, især fordi man ikke behøver at skrive så meget. Det gør det også lettere at ændre den nøgle, hvori melodien spilles – det vil sige, hvor højt eller lavt tonelejet ligger.

Hvis du programmerer på denne måde, skal du sætte talpar for tonelejet og varigheden (for eksempel **319,v*2** for den første tone) i **DATA** sætninger i begyndelsen af programmet.

Tæl, hvor mange toner der i alt er. Fortæl herefter maskinen:

FOR node=1 TO hvor mange der nu er
READ tone, varighed
SOUND 1,tone,varighed
NEXT node

Forbedringer og variationer

A. Melodien lyder sikkert lidt forkert, fordi der ingen pauser er mellem tonerne. Hvis to toner i det samme toneleje spilles lige efter hinanden, smelter de sammen og lyder som en enkelt tone. Det problem kan du løse ved at bruge:

SOUND 1,Ø,5

for at sætte en kort pause ind, hvor det er nødvendigt.

B. Du kunne få teksten til at stå på skærmen, mens melodien bliver spillet ved at bruge **PRINT.** Det kunne også hjælpe dig med at se, hvor i melodien, du er kommet til, når du vil rette i programmet.

Opgave 5 – Løver og antiloper

Meningen med dette program er at skabe en situation, der ligner den, en opsynsmand i en afrikansk dyrepark befinder sig i. Vi skal se på løverne og antiloperne i dyreparken. Løverne tager antiloperne som bytte (cirka 2ØØ om året), og opsynsmanden skal så beslutte, hvor mange løver han skal nedlægge for at holde bestanden under kontrol. Hovedtrækkene af programmet ser således ud:

1. Brug **PRINT** sætninger til at lade maskinen forklare situationen for en, der ikke kender programmet.

2. Sæt startværdierne for antallet af løver og antiloper. Det ville være rimeligt at starte med cirka 15 løver og 1ØØØØ antiloper.

3. Få maskinen til at spørge efter det antal løver, der skal nedlægges dette år og giv svaret i en **INPUT** sætning. Dette antal kaldes nedlagte.

4. Beregn antallet af løver og antiloper i det følgende år ved at bruge ligningerne herunder:

løver=INT(løver–nedlagte+løver*(antiloper–4ØØ*løver)/1ØØØØ)

**antiloper=INT(antiloper–2ØØ*løver+antiloper*
(2ØØØØ–antiloper)/3ØØØØ)**

Ligningen for løverne beregner det nye antal løver ud fra forrige års bestand, minus antallet af nedlagte, plus et antal fødte og døde afhængigt af, hvor mange løver der er til at avle unger og hvor mange antiloper, de kan nedlægge som bytte. Ligningen for antiloper er forrige års bestand, minus løvernes bytte, plus et antal fødte og døde. Få maskinen til at udskrive det nye antal løver og antiloper i en **PRINT** sætning.

5. Få maskinen til at spørge, om du ønsker programmet udført igen, og hvis ja, gå tilbage til punkt 3.

Forbedringer og variationer

A. Prøv at ændre startværdierne for løver og antiloper enten ved at bruge en **INPUT** sætning, således at brugeren bestemmer dem eller ved at bruge **INT** og **RND** og således lade maskinen vælge dem.

B. Få maskinen til at holde styr på antallet af år og lad den udskrive, hvilket år man er kommet til for hver gang, programmet udføres. I stedet for at spørge, om du ønsker at fortsætte efter hvert år, kunne den så gøre dette efter hvert tiende år.

C. Skriv programmet således, at det kun bliver udført, hvis der stadig er løver og antiloper i live.

D. Indfør nogle sætninger i programmet, der lykønsker brugeren, hvis bestanden af løver og antiloper er vokset. Hvis bestanden falder meget, skal brugeren irettesættes.

E. Lav skærmbilledet mere overskueligt ved at bruge **CLS** (til at viske skærmen ren), **PRINT** uden noget efter (for at få blanke linjer).
Man kan også sætte ekstra mellemrum mellem ordene for ikke at skulle dele et ord mellem to linjer.

F. Hvis du har mod på det, kunne du forsøge at tegne en plan over parken og bruge nogle symboler for løverne og antiloperne (for eksempel ét symbol for løverne og et andet for 1ØØ antiloper).

Andre ideer til programmer

1. Et hasardspil, der benytter tilfældige tal.
2. En staveprøve.
3. En række spørgsmål om et bestemt emne.
4. En musikskala.
5. Et spørgeskema.
6. Et farvediagram.
7. Et videnskabeligt eksperiment.
8. Et ur.
9. Et pengeskab.
10. Et spil, som laver nye ord ved at flytte rundt på bogstaverne i et enkelt ord.
11. En oversætter mellem to sprog.
12. Sætninger, hvor ord, der mangler, skal indsættes.
13. Et program, der skaber mønstre.
14. En liste med navne og adresser.
15. Tegning af en plan med angivelse af mål.
16. En ordleg.
17. Et program, der sorterer alfabetisk.
18. En talleg med spørgsmål og svar.
19. Et program, der veksler udenlandske penge.
20. En vittighedstegning.
21. En labyrint.
22. En rampe til affyring af raketter.
23. En biografreklame.
24. En spørgeleg.
25. En melodi.

Du bliver sikkert nødt til at gå tilbage og se på nogle af bogens eller dine egne programmer. Nogle gange kan du også hente hjælp til bestemte ting ved at bruge opslagsafsnittene.

OPSLAG

FR. B. O. GORM

Mine råd

Kontrollér altid betydningen af de ord og udtryk, du bruger, på samme måde, som du slår fremmedord op i en ordbog. De fleste af ordene og udtrykkene er lette at forstå, men der er en masse af dem. Hvis du føler dig usikker, må du hellere kontrollere betydningen og brugen af dem. Alle de ord og udtryk, du sandsynligvis får brug for, står i otte opslagsafsnit. Derfor skal du først finde ud af, hvilket afsnit det ord, du søger, står under og derpå slå op på afsnittet.
Afsnittene er:
Editering og skrivning af programmer
Brug af skærmen
Frembringelse af toner
Ordfunktioner
Talfunktioner
Logiske udtryk
Lagring og indlæsning af programmer
Generelle ordrer og funktioner
Det vil være en hjælp for dig, hvis du også selv laver en oversigt over de udtryk, du støder på. Du behøver ikke at inddele dem i afsnit, men det bliver lettere at slå op, hvis du skriver udtrykkene i alfabetisk orden.

Bodil Oda Gorm

Henvisninger

Du kan finde alle de udtryk, der hidtil er brugt i bogen, i opslagsafsnittene. Men her står også nogle nye udtryk. De er taget med, fordi de er gode at kende, når du skriver dine egne programmer. Hvis du vil skrive et program, der behandler ord, kan du få megen brug for nogle af ordfunktionerne. I et matematikprogram er det talfunktionerne, du skal anvende.
Opslagsafsnittene giver kun en kort oversigt over betydningen og brugen af udtrykkene, så hvis der er noget, du vil have uddybet, må du slå det op i andre bøger. Den mest nærliggende bog er:

Amstrad CPC 464 Bruger-vejledning

som følger med datamaskinen.

Der er også en del datablade, som kan være nyttige at læse. Her kan du ofte finde interessante programmer, som kan tilpasses dine ideer. På udenlandsk, især engelsk, findes der en mængde blade, men der findes også danske datablade, nemlig:

Alt om Data. Forlaget Ny Elektronik.
Mikro-bladet. Forlaget Mikro-bladet.
Soft. Forlaget Ny Elektronik.

Af danske datablade er det mest relevante:
Amstradbladet. MICROTECH Hovedgårdsvej 4, 8600 Silkeborg.

Udover dette, findes der en Amstrad Brugerklub, som kan kontaktes på adressen:
Amstrad Bruger Center
Jagtvej 201
2100 København Ø.

Her er en liste over alle de udtryk, Amstrad CPC 464 forstår. De udtryk, der er fremhævet, er dem, du sikkert oftest får brug for. De bliver beskrevet nærmere på de følgende sider. Vent med at bruge de andre udtryk, til du har lært mere om BASIC.

Editering og skrivning af programmer

AUTO
CONT
DELETE
EDIT
ERL
ERR
ERROR
FRE
HIMEM
LIST
MEMORY
NEW
ON ERROR GOTO
REM
RENUM
RESUME
RUN
STOP
TRON
TROFF

Brug af skærmen

BORDER
CLG
CLS
DRAW
DRAWR
INK
LOCATE
MODE
MOVER
ORIGIN
PAPER
PEN
PLOT
PLOTR
POS
PRINT
SPEED INK
SYMBOL
SYMBOL AFTER
TAB
TAG
TAGOFF

TEST
TESTR
VPOS
WINDOW
WINDOW SWAP
WRITE
XPOS
YPOS
ZONE

Frembringelse af toner

ENT
ENV
ON SQ GOSUB
RELEASE
SOUND
SQ

Ordfunktioner

ASC
BIN$
CHR$
INKEY$
INSTR
LEFT$
LEN
LOWER$
MID$
RIGHT$
SPACE$
STR$
STRING$
UPPER$
VAL

Talfunktioner

ABS
ATN
CINT
COS
CREAL
DEG
EXP
FIX
HEX$
INKEY
INT
LOG
LOG1Ø
MAX
MIN
MOD
PI
RAD
RANDOMIZE
RND
ROUND
SGN
SIN
SQR
TAN
UNT

Logiske udtryk

AND
NOT
OR
XOR

Lagring og indlæsning af programmer

CAT
CHAIN
CHAIN MERGE
CLOSEIN
CLOSEOUT
EOF
LOAD
MERGE
OPENIN
OPENOUT
SAVE
SPEED WRITE

Generelle ordrer og funktioner

AFTER
CALL
CLEAR
DATA
DEF FN
DEFINT
DEFREAL
DEFSTR
DI
DIM
EI
END
ERASE
EVERY
FOR/TO/STEP
GOSUB
GOTO
IF/THEN/ELSE
INP
INPUT
JOY
KEY
KEY DEF
LET
LINE INPUT
NEXT
ON/BREAK
ON/GOSUB
ON/GOTO
OUT
PEEK
POKE
READ
REMAIN
RESTORE
RETURN
SPEED KEY
TIME
WAIT
WEND
WHILE
WIDTH

Editering og skrivning af programmer

AUTO sætter automatisk linjenumre først i linjerne. Det kan afbrydes ved at trykke på **ESC**.
AUTO giver linjenumrene 1Ø, 2Ø, 3Ø ... o.s.v.
AUTO 12Ø giver linjenumrene 12Ø, 13Ø, 14Ø ... o.s.v.
AUTO 12Ø,4 giver linjenumrene 12Ø, 124, 128 ... o.s.v.
AUTO ,2 giver linjenumrene 1Ø, 12, 14, 16 ... o.s.v.

Amstrad Bruger-vejledning kap. 8, s. 4.

CONT bruges til at fortsætte udførelsen af et program fra det sted, hvor den blev afbrudt af **ESC** (der var blevet trykket ned to gange) eller af **STOP** eller **END** i selve programmet.

Amstrad Bruger-vejledning kap. 8, s. 8.

DELETE sletter linjer i et program.
DELETE 22–3Ø sletter alle linjer fra og med 22 og til og med 3Ø.
DELETE –3Ø sletter alle linjer indtil (og inklusiv) linje 3Ø.
DELETE 3Ø– sletter alle linjer fra og med linje 3Ø.
DELETE 4Ø sletter linje 4Ø. Dette kunne også gøres ved at skrive 4Ø og trykke på **ENTER**.

Amstrad Bruger-vejledning kap. 8, s. 11.

EDIT anvendes til at ændre i programlinjer.
EDIT 4Ø udskriver linje 4Ø på skærmen med markøren stående over det første tegn på linjen. Ændringer kan så foretages ved at flytte markøren hen ad linjen ved hjælp af tasterne med pile på. På den måde indsættes eller slettes tegn ved brug af **DEL** og **CLR** tasterne.

Amstrad Bruger-vejledning kap. 1, s. 16.

LIST udskriver et program (se side 48).
LIST udskriver hele programmet.
LIST 1Ø udskriver kun linje 1Ø.
LIST –5Ø udskriver alle linjer til og med linje 5Ø.
LIST 5Ø– udskriver alle linjer fra og med linje 5Ø og til sidste programlinje.
LIST 5Ø–1ØØ udskriver alle linjer fra og med linje 5Ø og til og med linje 1ØØ.

Amstrad Bruger-vejledning kap. 8, s. 24.

NEW sletter det program, der eventuelt ligger i maskinens hukommelse. Herefter er maskinen klar til indtastning af et nyt program (se side 50).

Amstrad Bruger-vejledning kap. 8, s. 28.

REM er beregnet til at sætte en kommentar eller en bemærkning ind i et program, som hjælp til at huske, hvad programmet udfører. Maskinen overser alt, hvad der står på linjen efter **REM**.
4Ø REM Maskinen vil overse dette.

Amstrad Bruger-vejledning kap.8, s.38.

RENUM omnummererer programlinjer.
RENUM omnummererer linjerne 1Ø, 2Ø, 3Ø ... o.s.v.
RENUM 1ØØ omnummererer linjerne 1ØØ, 11Ø, 12Ø ... o.s.v.
RENUM 1ØØ,5Ø lader alle linjenumre før linje 5Ø stå, som de er, men omnummererer linjerne herefter 1ØØ, 11Ø, 12Ø ... o.s.v.
RENUM 1ØØ,5Ø,5 lader alle linjenumre før linje 5Ø stå, som de er, men omnummererer linjerne herefter 1ØØ, 1Ø5, 11Ø ... o.s.v.
RENUM 1ØØ,,5 omnummererer linjerne 1ØØ, 1Ø5, 11Ø ... o.s.v.

Amstrad Bruger-vejledning kap.8, s.39.

RUN giver maskinen besked på at udføre et program (se side 50).
RUN 1ØØ giver maskinen besked på at udføre et program fra linje 1ØØ.
RUN "navn" får maskinen til at indlæse et program ved navn "navn" fra kassettebåndet og herpå udføre det.

Amstrad Bruger-vejledning kap.8, s.41.

STOP afbryder udførelsen af et program. Udførelsen kan sættes i gang igen med **CONT**.

Amstrad Bruger-vejledning kap.8, s.45.

TRON får maskinen til at udskrive linjenumrene linje for linje i den rækkefølge, de udføres i programmet. Dette kan være en hjælp, når man forsøger at finde nogle fejl i et program. For at stoppe udskrivningen skal man bruge **TROFF**.

Amstrad Bruger-vejledning kap.8, s.49.

Brug af skærmen

BORDER anvendes til at ændre farven på kanten af skærmbilledet (se side 42).
BORDER 9 giver en grøn kant.
BORDER 9,1Ø giver en kant, der blinker skiftende mellem farverne grøn og lyseblå.

Amstrad Bruger-vejledning s.G3.2.

CLG sletter det grafiske skærmbillede. Det grafiske skærmbillede er normalt hele skærmbilledet inden for kanten, men kan ændres ved brug af **ORIGIN**.

CLG visker den grafiske skærm ren og sætter baggrundsfarven til den farve, der sidst er blevet bestemt i en **CLG** ordre eller til ink Ø, hvis ingen er blevet specielt bestemt.
CLG 2 visker den grafiske skærm ren og sætter baggrundsfarven til farve 2.

Amstrad Bruger-vejledning kap.8, s.7.

CLS sletter tekstskærmen. Det er normalt hele skærmbilledet inden for kanten, men man kan have helt op til 8 tekstvinduer ved brug af **WINDOW** ordren.
CLS sletter tekstvindue Ø (hele skærmbilledet, medmindre man har bestemt det på anden måde) og sætter baggrundsfarven til den aktuelle PAPER-farve.
CLS 3 sletter tekstvindue 3 og sætter baggrundsfarven til den aktuelle PAPER-farve.

Amstrad Bruger-vejledning kap.8, s.8.

DRAW tegner en linje på skærmen (se side 38).
DRAW 2ØØ,3ØØ tegner en linje fra punktet Ø,Ø til punktet 2ØØ,3ØØ. Enhver yderligere ordre tegner linjen videre fra punktet 2ØØ,3ØØ til det næste punkt. Det første tal efter **DRAW** står for den vandrette afstand hen over skærmen, og det andet tal betegner den lodrette afstand. Punktet Ø,Ø er i det nederste venstre hjørne, medmindre det er ændret ved brug af **ORIGIN** ordren.
DRAW 2ØØ,3ØØ,13 tegner en linje til punktet 2ØØ,3ØØ med den farve, der aktuelt bestemmes af 13.

Amstrad Bruger-vejledning kap.8, s.12.

DRAWR virker på samme måde som **DRAW** med undtagelse af, at den tegner ud fra det sted, hvor den grafiske markør står.
DRAWR 2Ø,3Ø,13 tegner en linje med farven 13 til et punkt 2Ø enheder vandret og 3Ø enheder lodret fra det punkt, man er ved.

Amstrad Bruger-vejledning kap.8, s.12.

INK bruges til at vælge den farveskala, der skal benyttes til pen, paper og grafik (se side 44).
INK 1,8 giver paper 1 og pen 1 farven lyslilla.
INK 1,7,8 får farven på paper 1 og pen 1 til at blinke skiftende mellem purpur og lyslilla.

Amstrad Bruger-vejledning s.G3.2 og G3.5.

LOCATE bruges til at flytte tekstmarkøren til et nyt punkt på skærmen. Teksten vil da blive udskrevet fra dette punkt (se side 53).
LOCATE 2Ø,1Ø flytter tekstmarkøren til et punkt 2Ø enheder fra den venstre margen og 1Ø enheder fra den øverste kant af skærmbilledet.
LOCATE 2,2Ø,1Ø flytter tekstmarkøren 2Ø enheder vandret og 1Ø enheder lodret fra det øverste venstre hjørne af tekstvindue 2.

Amstrad Bruger-vejledning kap.8, s.25.

MODE anvendes til at vælge den tilstand, grafik og tekst vises i på skærmen (se side 37).
MODE Ø kan give op til 16 farver, 2Ø tegn pr. linje og i alt 25 linjer på skærmen.
MODE 1 kan give op til 4 farver, 4Ø tegn pr. linje og i alt 25 linjer på skærmen.
MODE 2 giver 2 farver, 8Ø tegn pr. linje og 25 linjer på skærmen.

Amstrad Bruger-vejledning kap. 5, s. 3.

MOVE flytter den grafiske markør til et punkt på skærmen uden at tegne på vej derhen (se side 39).
MOVE 2ØØ,3ØØ flytter markøren til punktet 2ØØ,3ØØ fra punktet Ø,Ø. Det første tal efter MOVE betegner den vandrette afstand hen over skærmen, og det andet tal står for den lodrette afstand. Punktet Ø,Ø er det nederste venstre hjørne af skærmen, medmindre det er ændret ved **ORIGIN** ordren.

Amstrad Bruger-vejledning kap. 8, s. 28.

MOVER virker på samme måde som **MOVE** med undtagelse af, at den flytter den grafiske markør i forhold til dens aktuelle position.
MOVER 2Ø,3Ø flytter markøren til et punkt 2Ø enheder til højre for og 3Ø enheder op fra det punkt, man sidst var ved.

Amstrad Bruger-vejledning kap. 8, s. 28.

ORIGIN bruges til at vælge et udgangspunkt for den grafiske markør.
ORIGIN kan også bruges til at bestemme et grafisk vindue.
ORIGIN 32Ø,2ØØ placerer punktet Ø,Ø i midten af skærmbilledet.
ORIGIN Ø,Ø,4Ø,6ØØ,36Ø,4Ø bestemmer et grafisk vindue 4Ø enheder fra kanten i alle retninger.

Amstrad Bruger-vejledning kap. 8, s. 32.

PAPER bruges til at vælge baggrundsfarven på skærmbilledet inden for kanten (se side 43).
PAPER 1 sætter baggrundsfarven til den farve, der aktuelt bestemmes af 1.
PAPER 3,1 sætter baggrundsfarven i vindue 3 til den farve, der aktuelt bestemmes af 1.

Amstrad Bruger-vejledning s. G3.2 og kap. 8, s. 33.

PEN vælger den farve, der bliver skrevet med på skærmen (se side 43).
PEN 3 udskriver tegnene med den farve, der aktuelt bestemmes af tallet 3 (det er rødt i tilstandene Ø og 1, medmindre dette er ændret ved en **INK** ordre).
PEN#4,3 udskriver tegnene med farven 3 i tekstvindue 4.

Amstrad Bruger-vejledning s. G3.2 og kap. 8, s. 34.

PLOT Benyttes til at afmærke enkelte punkter på skærmen (se side 40).
PLOT 2ØØ,3ØØ fremhæver punktet 2ØØ enheder til højre for og 3ØØ enheder oven over punktet Ø,Ø. Punktet Ø,Ø er det nederste venstre hjørne af skærmen, medmindre dette er ændret ved en **ORIGIN** ordre.
PLOT 2ØØ,3ØØ,2 fremhæver punktet 2ØØ,3ØØ med farven 2.

Amstrad Bruger-vejledning kap. 8, s. 35.

PLOTR virker på samme måde som **PLOT** med undtagelse af, at den fungerer ud fra den grafiske markørs aktuelle position.
PLOTR 2Ø,3Ø,2 fremhæver med farven 2 et punkt 2Ø enheder til højre for og 3Ø enheder oven over det punkt, man sidst var ved.

Amstrad Bruger-vejledning kap. 8, s. 35.

PRINT bruges til at udskrive tegn på skærmen (se side 23).
PRINT 5,6 giver 5 **6**
PRINT 5;6 giver 5 **6**
PRINT 5,,,6 giver 5
6
PRINT "ord" udskriver **ord**.
PRINT #3,"Goddag" udskriver **Goddag** i vindue 3.

Amstrad Bruger-vejledning kap. 3, s. 4.

TAB benyttes til at udskrive fra en bestemt række (lodret) på skærmen (se side 52).
PRINT TAB(9) "Goddag" udskriver **Goddag** med start i den niende lodrette række på skærmen.

Amstrad Bruger-vejledning kap. 3, s. 6.

TEST fortæller farven på et punkt på skærmen.
PRINT TEST(35Ø,1ØØ) udskriver det tal, som bestemmer farven på punktet 35Ø,1ØØ.

Amstrad Bruger-vejledning kap. 8, s. 48.

TESTR virker på samme måde som **TEST** med undtagelse af, at den fortæller farvenummeret på et punkt set i forhold til den grafiske markørs aktuelle position.
TESTR(3Ø,2Ø) giver farvenummeret på punktet 3Ø enheder til højre for og 2Ø enheder oven over det punkt, man sidst var ved.

Amstrad Bruger-vejledning kap.8, s.48.

WINDOW bruges til at inddele skærmen i forskellige områder, der kaldes vinduer. Man kan tildele vinduerne forskellig baggrundsfarve og få udskrevet tegn i det vindue, man bestemmer i en **PRINT** sætning.

En **WINDOW** ordre består af **WINDOW** # vinduesnummer, venstre række, højre række, øverste række, nederste række.

```
1Ø MODE 1
2Ø WINDOW #3,7,17,7,12
3Ø PAPER #3,2
4Ø PEN #3,3
5Ø CLS #3
6Ø PRINT #3, "Goddag"
```

Amstrad Bruger-vejledning kap.5, s.10.

Frembringelse af toner

SOUND benyttes til at spille toner (se side 45).
SOUND 1,478 spiller nøgle C'et (toneperiode 478) på kanal 1. Datamaskinen antager, at man vil have tonen spillet i en femtedel af et sekund med middel styrke.
SOUND 1,478,1ØØ spiller den samme tone i et sekund (1ØØ gange Ø.Ø1 sekund).
SOUND 1,478,1ØØ,7 spiller den samme tone med højeste styrke (7 ud af 7 mulige).

Amstrad Bruger-vejledning s.G3.14.

Ordfunktioner

INKEY$ kontrollerer tastaturet og ser, om der er blevet trykket på en tast.

```
1Ø a$=INKEY$
2Ø IF a$="" GOTO 1Ø
3Ø IF a$="y" THEN PRINT "ja":END
4Ø IF a$="n" THEN PRINT "nej" ELSE GOTO 1Ø
```

Amstrad Bruger-vejledning kap.8, s.20.

INSTR søger gennem en streng (en række tegn) for at se, om den indeholder en anden streng.
x=INSTR (a$,b$) ser om det første tegn i **b$** findes i **a$** og gemmer svaret i **x**.
x=INSTR (n,a$,b$) gør det samme, men starter her søgningen fra den n'de position.
PRINT INSTR("MOGENS","G") udskriver **3**.
PRINT INSTR("MOGENS","S") udskriver **6**.
PRINT INSTR(4,"DRENGE","E") udskriver **6**.

Amstrad Bruger-vejledning kap. 8, s. 22.

LEFT$ kopierer den venstre del af et ord (se side 62).

```
1Ø a$="DRENG"
2Ø b$=LEFT$(a$,3)
3Ø PRINT b$
```

udskriver **DRE**.

Amstrad Bruger-vejledning kap. 8,s. 23.

LEN giver antallet af tegn i et ord (se side 64).

```
1Ø K=LEN("SVENDBORG")
2Ø PRINT K
```

udskriver **9**.

Amstrad Bruger-vejledning kap. 8, s. 24.

LOWER$ ændrer bogstaverne i et ord fra store til små bogstaver.
PRINT LOWER$("GODDAG") udskriver **goddag.**

Amstrad Bruger-vejledning kap. 8, s. 26.

MID$ kopierer den miderste del af et ord (se side 62).

```
1Ø a$="SUSANNE"
2Ø b$=MID$(a$,2,3)
3Ø PRINT b$
```

udskriver **SAN**.

Amstrad Bruger-vejledning kap. 8, s. 27.

RIGHT$ kopierer den højre del af et ord (se side 62).

```
1Ø a$="WIENERBRØD"
2Ø b$=RIGHT$(a$,4)
3Ø PRINT b$
```

udskriver **BRØD**.

Amstrad Bruger-vejledning kap. 8, s. 40.

SPACE$ udskriver mellemrum.
PRINT SPACE$(12) udskriver en række af 12 mellemrum.

Amstrad Bruger-vejledning kap. 8, s. 43.

STRING$ gentager et tegn et bestemt antal gange.
PRINT STRING$(7,"*") udskriver *******.

Amstrad Bruger-vejledning kap. 8, s. 46.

UPPER$ ændrer bogstaverne i et ord fra små til store bogstaver.
PRINT UPPER$("goddag") udskriver **GODDAG**.

Amstrad Bruger-vejledning kap. 8, s. 49.

Talfunktioner

ABS giver et tals positive værdi.
PRINT ABS(–7) udskriver **7**.

Amstrad Bruger-vejledning kap. 8, s. 3.

ATN giver vinklen, hvis tangent kendes. Resultatet bliver i radianer, medmindre det er angivet på anden måde med en **DEG** ordre.

Amstrad Bruger-vejledning kap. 8, s. 4.

CINT afrunder et tal til det nærmeste hele tal.
PRINT CINT(35.76) giver **36**.

Amstrad Bruger-vejledning kap. 8, s. 6.

COS giver cosinus til en vinkel. Datamaskinen antager, at målene skal opgives i radianer, medmindre dette er angivet anderledes i en **DEG** ordre.

Amstrad Bruger-vejledning kap. 8, s. 8.

DEG giver maskinen besked på, at den skal udregne funktionsværdier som **COS** i grader i stedet for radianer.

Amstrad Bruger-vejledning kap. 8, s. 10.

EXP giver e (omtrent 2.7183) til den angivne potens.

Amstrad Bruger-vejledning kap. 8, s. 17.

INT giver heltalsdelen af et tal (se side 62).
INT(45.61) giver **45**.
INT(–17.22) giver **–18**.

Amstrad Bruger-vejledning kap. 8, s. 22.

LOG opgiver den naturlige logaritme til et tal.

Amstrad Bruger-vejledning kap. 8, s. 25.

LOG1Ø giver titalslogaritmen til et tal.

Amstrad Bruger-vejledning kap.8, s.26.

MOD opgiver den rest, der bliver tilbage ved en division.
PRINT 11 MOD 4 giver **3**.
PRINT 14 MOD 3 giver **2**.

PI er en konstant. Dens værdi er 3.14159265. En cirkels omkreds med radius **r** er **2*PI*r.**

Amstrad Bruger-vejledning kap.8, s.34.

RAD giver maskinen besked på at udregne værdien af funktioner som COS i radianer. RAD ophæver **DEG** ordren.

Amstrad Bruger-vejledning kap.8, s.37.

RND finder et tilfældigt tal mellem Ø og 1 (men ikke 1). Den skal bruges sammen med **INT** for at finde hele tal (se som eksempel siderne 62 og 63).

Amstrad Bruger-vejledning kap.8, s.40.

SGN opgiver et tals fortegn.
1 for positive
–1 for negative
Ø for nul.

Amstrad Bruger-vejledning kap.8, s.42.

SIN angiver sinus til en vinkel. Datamaskinen antager, at målene skal opgives i radianer, med mindre dette er bestemt på anden måde i en **DEG** ordre.

Amstrad Bruger-vejledning kap.8, s.42.

SQR giver et tals kvardratrod.
PRINT SQR(16) udskriver **4.**

Amstrad Bruger-vejledning kap.8, s.45.

TAN angiver tangenten til en vinkel. Maskinen opgiver målene i radianer, hvis dette ikke er anderledes bestemt i en **DEG** ordre.

Amstrad Bruger-vejledning kap.8, s.48.

Logiske udtryk

AND benyttes til at kontrollere, at to betingelser opfyldes (se side 69).
IF x>Ø AND x<1Ø THEN PRINT "ciffer".

Amstrad Bruger-vejledning kap.4, s.18.

NOT anvendes sammen med **IF** og **THEN** til at kontrollere, at en betingelse ikke opfyldes.
IF NOT a=2 THEN PRINT "nej" udskriver **nej,** hvis **a** ikke er lig **2.**

Amstrad Bruger-vejledning kap.4, s.18.

OR anvendes til kontrol af, at en ud af to betingelser opfyldes (se side 71).
IF a=3 OR b=6 THEN PRINT "ja" udskriver **ja,** hvis enten **a=3** eller **b=6** (eller både **a=3** og **b=6**).

Amstrad Bruger-vejledning kap.4, s.18.

XOR bruges til at kontrollere, at en ud af to betingelser opfyldes (men ikke begge).
IF person1=pige XOR person2=pige THEN PRINT "ægteskab tilladt".

Amstrad Bruger-vejledning kap.8, s.18.

Lagring og indlæsning af programmer

CAT bruges til at se en oversigt over indholdet af et kassettebånd.

Amstrad Bruger-vejledning kap.2, s.8.

LOAD indlæser et program fra et kassettebånd.
LOAD "VAND" indlæser et program ved navn **VAND** i maskinens hukommelse, men programmet bliver ikke udført: sammenlign med **RUN** på side 86.

Amstrad Bruger-vejledning kap.8, s.25.

SAVE gemmer et program på kassettebåndet.
SAVE "RAKET" lagrer et program og giver det navnet **RAKET.**

Amstrad Bruger-vejledning kap.2, s.6.

SPEED WRITE anvendes til at angive den hastighed med hvilken, programmer og data lagres på kassettebåndet. Datamaskinen vil normalt lagre data med en hastighed på 1ØØØ signalenheder pr. sekund, men **SPEED WRITE 1** ændrer lagringshastigheden til 2ØØØ signalenheder pr. sekund. Jo større hastighed, jo mindre pålidelig overførelse.

Amstrad Bruger-vejledning kap.2, s.5.

Generelle ordrer og funktioner

CLEAR nulstiller alle de variable, der er brugt i et program.

Amstrad Bruger-vejledning kap.8, s.7.

DATA anvendt sammen med **READ** forsyner et program med data (se side 61).
DATA 1,2,3,4,5
DATA æble,banan,gulerod,daddel

Amstrad Bruger-vejledning kap.4, s.7.

DEF FN bruges til at bestemme en funktion.

Amstrad Bruger-vejledning kap.8, s.10.

DIM reserverer plads til en gruppe ord eller tal således, at de kan lægges ind i hukommelsen.
DIM x(2Ø) tillader, at tal lægges ind i maskinen på adresserne **x(Ø), x(1), x(2) ... x(2Ø).**
DIM x$(2Ø) tillader, at ord lægges ind i maskinen på adresserne **x$(Ø), x$(1), x$(2) ... x$(2Ø).**

Amstrad Bruger-vejledning kap.4, s.13.

END giver maskinen besked om, at et program er færdigt.

Amstrad Bruger-vejledning kap.8, s.13.

FOR/TO/STEP bruges sammen med **NEXT** til at gentage de samme programlinjer flere gange (se side 6Ø).

Amstrad Bruger-vejledning kap.8, s.18.

GOSUB er en programordre, der siger, at maskinen skal hoppe til en subroutine, der starter ved et givet linjenummer.
GOSUB 55Ø giver programmet besked på at hoppe til et underprogram, der starter på linje 55Ø.
RETURN anvendes til at markere, at et underprogram er slut.

Amstrad Bruger-vejledning kap.8, s.18.

GOTO er en programordre, der betyder, at programmet skal hoppe til et bestemt linjenummer, idet det springer alle de mellemliggende linjer over (se side 67).
GOTO 41Ø giver programmet besked på at hoppe til linje 41Ø.

Amstrad Bruger-vejledning s.G2.5.

IF/THEN/ELSE fortæller maskinen, at den skal udføre forskellige ting, der afhænger af, om forskellige betingelser opfyldes (se side 65).
IF a$="god" THEN GOTO 6ØØ ELSE GOTO 75Ø.

Amstrad Bruger-vejledning kap. 8, s. 19.

INPUT muliggør, at ord eller tal kan lægges ind i maskinen (se side 56).
INPUT a til tal.
INPUT a$ til ord.

Amstrad Bruger-vejledning s. G2.6.

NEXT anvendes sammen med **FOR/TO/STEP**.

Amstrad Bruger-vejledning kap. 8, s. 29.

ON/GOSUB og **ON/GOTO** bruges til at vælge mellem forskellige muligheder.
ON x GOSUB 1ØØ,2ØØ,3ØØ betyder, at hvis **x=1** skal programmet hoppe til underprogrammet, der starter på linje 1ØØ, hvis **x=2** skal det hoppe til underprogrammet med start på linje 2ØØ o.s.v.
ON x GOTO 1ØØ,2ØØ,3ØØ virker på samme måde.

Amstrad Bruger-vejledning kap. 8, s. 29.

READ anvendt med **DATA** læser information (se side 61).
1Ø READ A$,B$,C$,D$,E$
2Ø DATA rød,grøn,orange,blå,gul
3Ø PRINT B$
udskriver **grøn**.

Amstrad Bruger-vejledning kap. 4, s. 14.

RETURN bruges sammen med **GOSUB** som besked på at vende tilbage til et hovedprogram.

Amstrad Bruger-vejledning kap. 8, s. 40.

WEND markerer enden på en løkke, der begynder med **WHILE** (se side 64).

Amstrad Bruger-vejledning kap. 8, s. 51.

WHILE anvendes sammen med **WEND** og laver en løkke, hvor maskinen gentagende gange udfører programlinjer inden for løkken, så længe en bestemt betingelse er opfyldt (se side 64).

Amstrad Bruger-vejledning kap. 8, s. 51.

Indtastning af danske karakterer

Med følgende program er det muligt at få danske karakterer på skærmen af din Amstrad.

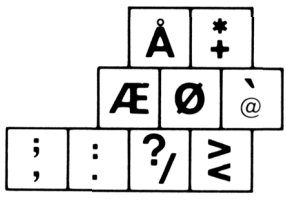
Programmet omdefinerer udseendet af [, \,], {, | og }.
Disse tegn bliver ændret til de danske bogstaver Æ, Ø, Å, æ, ø og å.

Foruden at omdefinere datamatens tegnsæt, bytter programmet også rundt på tasternes placering, således at tasterne vil være placeret som på en almindelig skrivemaskine.

Da datamaten ikke ved, at disse specialtegn skal tolkes som bogstaver, vil tasten **CAPS LOCK** ikke indvirke på disse danske tegn.
Programmet vil derfor spørge, om du ønsker store bogstaver som direkte tryk eller som shiftet tryk, svar derfor alt efter om du er vant til at skrive med store eller små bogstaver som standard.

Læg det indtastede program ud på et tomt kassettebånd, så du nemt kan finde programmet og køre det, hver gang du tænder din datamat.

```
 1Ø SYMBOL AFTER 91
 2Ø SYMBOL ASC ("["),126,216,216,254,216,216,222
 3Ø SYMBOL ASC ("\"),118,2ØØ,214,214,214,38,22Ø
 4Ø SYMBOL ASC ("]"),56,Ø,124,198,254,198,198
 5Ø SYMBOL ASC ("{"),Ø,Ø,116,26,126,216,11Ø
 6Ø SYMBOL ASC ("|"),Ø,Ø,118,204,214,1Ø2,22Ø
 7Ø SYMBOL ASC ("}"),48,Ø,12Ø,12,124,2Ø4,118
 8Ø INPUT "VIL DU HAVE STORE BOGSTAVER SOM
    DIREKTE TRYK ELLER SOM SHIFTET TRYK
    <D/S> ";A$
 9Ø AØ=ASC("["):BØ=ASC("\"):CØ=ASC("]")
1ØØ A1=ASC("{"):B1=ASC("|"):C1=ASC("}")
11Ø IF A$="S" THEN X=A1:A1=AØ:AØ=X:X=B1:
    B1=BØ:BØ=X:X=C1:C1=CØ:CØ=X
12Ø KEY DEF 29,1,AØ,A1
13Ø KEY DEF 28,1,BØ,B1
14Ø KEY DEF 26,1,CØ,C1
15Ø KEY DEF 22,1,ASC("<"),ASC(">")
16Ø KEY DEF 19,1,ASC("@"),ASC("`")
17Ø KEY DEF 17,1,ASC("+"),ASC("*")
18Ø KEY DEF 39,1,ASC(","),ASC(";")
19Ø KEY DEF 31,1,ASC("."),ASC(":")
```



brug din AMSTRAD

Med bogen lærer du at
– skrive dine egne BASIC-programmer
– tegne billeder på skærmen
– komponere musik
– programmere sjove spil

Udnyt din Amstrads mange muligheder uden i forvejen at kende noget som helst til datamater eller matematik. Du lærer det lynhurtigt.

ISBN 87-571-0889-7